

This electronic thesis or dissertation has been downloaded from the King's Research Portal at <https://kclpure.kcl.ac.uk/portal/>



From Index Locorum to Citation Network: an Approach to the Automatic Extraction of Canonical References and its Applications to the Study of Classical Texts

Romanello, Matteo

Awarding institution:
King's College London

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without proper acknowledgement.

END USER LICENCE AGREEMENT



Unless another licence is stated on the immediately following page this work is licensed

under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International

licence. <https://creativecommons.org/licenses/by-nc-nd/4.0/>

You are free to copy, distribute and transmit the work

Under the following conditions:

- Attribution: You must attribute the work in the manner specified by the author (but not in any way that suggests that they endorse you or your use of the work).
- Non Commercial: You may not use this work for commercial purposes.
- No Derivative Works - You may not alter, transform, or build upon this work.

Any of these conditions can be waived if you receive permission from the author. Your fair dealings and other rights are in no way affected by the above.

Take down policy

If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

FROM INDEX LOCORUM TO CITATION NETWORK:
AN APPROACH TO THE AUTOMATIC
EXTRACTION OF CANONICAL REFERENCES AND
ITS APPLICATIONS TO THE STUDY OF CLASSICAL
TEXTS

MATTEO ROMANELLO

Submitted for the degree of Ph.D.
School of Arts and Humanities
King's College London

May 2015

ABSTRACT

My research focusses on the automatic extraction of canonical references from publications in Classics. Such references are the standard way of citing classical texts and are found in great numbers throughout monographs, journal articles and commentaries.

In chapters 1 and 2 I argue for the importance of canonical citations and for the need to capture them automatically. Their importance and function is to signal text passages that are studied and discussed, often in relation to one another as can be seen in parallel passages found in modern commentaries. Scholars in the field have long been exploiting this kind of information by manually creating indexes of cited passages, the so-called *indices locorum*. However, the challenge we now face is find new ways of indexing and retrieving information contained in the growing volume of digital archives and libraries.

Chapters 3 and 4 look at how this problem can be tackled by translating the extraction of canonical citations into a computationally solvable problem. The approach I developed consists of treating the extraction of such citations as a problem of named entity extraction. This problem can be solved with some degree of accuracy by applying and adapting methods of Natural Language Processing. In this part of the dissertation I discuss the implementation of this approach as a working prototype and an evaluation of its performance.

Once canonical references have been extracted from texts, the web of relations between documents that they create can be represented as a network. This network can then be searched, manipulated, visualised and analysed in various ways. In chapter 5 I focus specifically on how this network can be leveraged to search through bodies of secondary literature. Finally in chapter 6 I discuss how my work opens up new research perspectives in terms of visualisation, analysis and the application of such automatically extracted citation networks.

ACKNOWLEDGEMENTS

Firstly, I would like to express my deepest gratitude to my supervisors, Willard McCarty and Shalom Lappin, who helped shaping my thinking and provided thoughtful guidance during their supervision. I would also like to thank Charlotte Roueché and Peter Heslin for their insightful comments on my dissertation and the enjoyable discussion at the Viva.

Many people in the Digital Classics community have been precious sources of guidance and inspiration, but special thanks go to: Gregory Crane and Neel Smith for having inspired my research with their work; Reinhard Förtsch for having been so supportive during the last four years; Eric Rebillard and Adam Chandlers for having generously shared research data with me and for their active support.

I am greatly indebted to Brett Lucas for having taught me the ins and outs of the English language and for the stimulating discussions we had. My sincere thanks go to Michele Pasin and Aurélien Berra who have read and commented on early chapter drafts, and to Agnes Thomas and Niels-Oliver Walkowski with whom I've shared ups and downs of postgraduate life.

I am also grateful to the friends and colleagues that I had the luck to meet in various places over the past six years for their advices and encouragement when they were needed the most: Alison Babeu at Tufts (where it all began); Gabby Bodard, Elena Pierazzo, Simon Mahony, Valeria Vitale and Alejandro Giacometti at the DDH in London; Francesco Mambrini, Felix Schäfer, Wolfgang Schmidle, Martina Trognitz, Fabian Körner and Elton Barker in Berlin.

Finally, I would like to thank my family and especially Rachele for her invaluable support and for always keeping me grounded.

CONTENTS

List of Figures	5
List of Tables	8
Nomenclature	10
1 INTRODUCTION	13
1.1 The Focus of this Thesis: the Automatic Extraction of Canonical References	13
1.2 Motivations for this Work	14
1.3 The Nature of this Research	15
1.4 Citations and Information Retrieval	16
1.4.1 Characteristics of References in the Classics	16
1.4.2 Information Retrieval in the Humanities	18
1.5 Research Aims and Contribution	21
1.6 Reader's Guide	22
2 CANONICAL REFERENCES AND PARALLEL PASSAGES	25
2.1 Citing Parallel Passages: a Diachronic Perspective	25
2.1.1 Selected Examples of Classical Commentaries	26
2.1.2 XVII th century	27
2.1.3 XIX th century	28
2.1.4 XX th century	30
2.1.5 XXI st century	32
2.2 Retrieving Parallel Passages: Electronic Concordances	35
2.2.1 Classical Scholarship and Computer Technology	35
2.2.2 Tools Shape Research Questions	37
2.2.3 Reading, Remembering, Retrieving	40
2.3 Beyond Electronic Concordances	42
2.3.1 The Critical Apparatus of Digital Editions	43
2.3.2 Digital Editions of Quoted Texts	43
2.3.3 Tools for the Study of Intertextuality	44
2.4 Summary	48
3 DISENTANGLING CANONICAL CITATIONS	49

3.1	Transforming a Human-Readable Notation into a Formal Model	49
3.2	Understanding Canonical Texts	52
3.2.1	Canonical Citations and their Importance for a Scholarly Discourse About Texts	53
3.2.2	Canonical Citation Schemes as Historical Objects . . .	54
3.2.3	Main Characteristics of Canonical Citation Schemes .	59
3.3	Key Concepts of Ontologies and Semantic Web	60
3.3.1	What is an Ontology?	60
3.3.2	Methods for Building Ontologies	63
3.3.3	Publishing Ontologies and Semantic Data on the Web	64
3.4	Computational Models of Citations: a Review	68
3.4.1	Conceptual Models of Bibliographic Information . . .	69
3.4.2	Citations in Formal Ontologies	73
3.4.3	Citations in Digital Classics	76
3.5	A Formal Model of Citations: the HuCit Ontology	85
3.5.1	Methodology and Rationale	86
3.5.2	Overview of HuCit	88
3.5.3	Modelling a Citation and the Citing Document	89
3.5.4	Modelling the Structure of the Cited Text	93
3.5.5	Modelling Authority Data	97
3.6	A Knowledge Base to Support the Extraction of Citations .	100
3.6.1	Technical Implementation	100
3.6.2	Populating the Knowledge Base	101
3.6.3	Uses of the Knowledge Base	106
3.7	Summary	109
4	AUTOMATIC EXTRACTION OF CANONICAL CITATIONS	110
4.1	Key Concepts in Information Extraction	110
4.1.1	Extraction of Named Entities	110
4.1.2	Methods in Natural Language Processing	112
4.1.3	Evaluation Metrics: Precision, Recall and F_1 Score . .	113
4.2	The Landscape of Information Extraction Research	116
4.2.1	Information Extraction Systems	116
4.2.2	Citations and other Discipline-specific Named Entities	119
4.2.3	The Extraction of Modern Bibliographic References .	120
4.3	Creation of Annotated Datasets	121
4.3.1	A Scheme to Annotate Canonical Citations	121

4.3.2	The Datasets: APh and JSTOR	125
4.3.3	File Formats of the Datasets	131
4.4	The Information Extraction Pipeline	133
4.4.1	Pre-processing	134
4.4.2	Named Entity Extraction	136
4.4.3	Relation Detection	140
4.4.4	Entity and Relation Disambiguation	142
4.5	Evaluation of the Performance	149
4.5.1	Evaluation of Named Entity Extraction	151
4.5.2	Evaluation of Relation Detection	155
4.5.3	Evaluation of Entity and Relation Disambiguation	157
4.6	Discussion and Further Work	162
4.7	Summary	165
5	CITATION NETWORKS AND THE STUDY OF CLASSICAL TEXTS	167
5.1	From Index Locorum to Citation Network	167
5.1.1	Scale and Accuracy	168
5.1.2	Manipulability	169
5.1.3	Networks of Relations	169
5.2	Network Approaches to Citations	170
5.3	Texts through the Lens of a Network	173
5.3.1	A Three-Level Citation Network	174
5.3.2	Network-based Search	179
5.3.3	Advantages and Limitations	184
5.4	Summary	186
6	CONCLUSIONS	187
6.1	Overall Contributions of this Work	187
6.1.1	A System to Extract Canonical References	187
6.1.2	An Ontology of Canonical References	189
6.1.3	A Citation-based Search	189
6.2	Future Development	190
6.2.1	Improving the Accuracy of Canonical Reference Ex- traction	190
6.2.2	Extending the Extraction to other Kinds of References	191
6.2.3	Citation Extraction as Research Infrastructure	192
6.3	Future Use and Implications	193
6.3.1	Enhancing the Reading of Classical Texts	193
6.3.2	Mining Intertextual Parallels from Commentaries	194

6.3.3 Analysis and Visualisation of the Citation Network	196
6.4 Final Reflections	196
Glossary	198
Bibliography	200

LIST OF FIGURES

Figure 2.1	An excerpt of La Cerda's commentary on the <i>Aeneid</i> (1612)	28
Figure 2.2	An excerpt of Lachmann's commentary on Lucretius' <i>De Rerum Natura</i> (1850)	29
Figure 2.3	An excerpt of Jebb's commentary on Sophocles' <i>Electra</i> (1894)	31
Figure 2.4	An excerpt of Wilamowitz's commentary on Euripides' <i>Herakles</i> (1895)	32
Figure 2.5	An excerpt of Fraenkel's commentary on Aeschylus' <i>Agamemnon</i> (1950)	33
Figure 2.6	An excerpt of Gibson's commentary on book 3 of Ovid's <i>Ars Amatoria</i> (2003)	33
Figure 3.1	Graph diagram of a reference to Plato <i>Rep.</i> 595a-596a	51
Figure 3.2	The symbols used in this chapter to illustrate the structure of the ontology	61
Figure 3.3	Graph representation of an RDF triple.	66
Figure 3.4	An example of RDF triples expressed using the XML syntax.	67
Figure 3.5	Example of RDF triples expressed using the Turtle syntax.	67
Figure 3.6	Linking Open Data cloud diagram 2014	68
Figure 3.7	The FRBR hierarchy: modelling a copy of the <i>Iliad</i>	70
Figure 3.8	The key entities of the CIDOC CRM.	72
Figure 3.9	An example response from the CWKB resolution service	77
Figure 3.10	The syntax of a CTS URN.	79
Figure 3.11	The XML reply of the Perseus' CTS API upon a Get - Capabilities request	81
Figure 3.12	The XML reply of the Perseus' CTS API upon a Get - Passage request	82
Figure 3.13	The XML reply of the Perseus' CTS API upon a Get - ValidReff request	83

Figure 3.14	CWKB: the record for Vergil's <i>Eclogues</i> , expressed using the Turtle syntax.	85
Figure 3.15	CIDOC CRM: the class <code>E28_Conceptual_Object</code> . . .	89
Figure 3.16	An example abstract drawn from the APh.	90
Figure 3.17	HuCit ontology: classes <code>Document</code> , <code>Sentence</code> , <code>Citation</code> and <code>CanonicalCitation</code> and their relation to CIDOC CRM and FRBR _{OO}	91
Figure 3.18	HuCit ontology: modelling the content of an APh abstract	92
Figure 3.19	HuCit ontology: modeling the form and content of the citation <i>Aen.</i> 1,1-11	93
Figure 3.20	HuCit ontology: modelling the structure of the cited text	94
Figure 3.21	HuCit ontology: modelling the content of a canonical citation	95
Figure 3.22	HuCit ontology: assignment of a CTS URN to the corresponding <code>TextElement</code> instance	96
Figure 3.23	HuCit ontology: usage of class <code>E55_Type</code> to define a typology of <code>TextElement</code> instances	97
Figure 3.24	HuCit ontology: the classes used to model authority data	98
Figure 3.25	HuCit ontology: modelling the authority data related to Vergil	99
Figure 3.26	HuCit ontology: modelling the authorship of the <i>Aeneid</i>	100
Figure 3.27	Knowledge base: the record for Vergil expressed as Turtle RDF	102
Figure 3.28	The XML reply of the Perseus' CTS API upon a <code>Get-Capabilities</code> request	104
Figure 3.29	Knowledge base: the instances corresponding to lines 1-2 of <i>Aeneid</i> , book 1 expressed as Turtle RDF.	105
Figure 4.1	The four error types used in the evaluation of named entity extraction	114
Figure 4.2	The four error types and their relation to computing precision and recall	115
Figure 4.3	An example of annotated APh document visualised in Brat	122

Figure 4.4	An example of discursive canonical reference annotated and visualised in Brat	124
Figure 4.5	The disambiguation of a scope relation visualised in Brat	124
Figure 4.6	The annotation of a canonical reference represented in the IOB format	131
Figure 4.7	The annotation of a canonical reference represented as standoff markup	132
Figure 4.8	Overview of the information extraction pipeline. . .	133
Figure 4.9	An example of the pipeline input	134
Figure 4.10	The execution of the rule-based algorithm to detect scope relations	142
Figure 4.11	The grammar rules used to tokenise a citation scope	148
Figure 4.12	The grammar rules used to parse a citation scope . .	149
Figure 4.13	The result of parsing a citation scope represented as parse tree and JSON object.	150
Figure 4.14	An example of scope relations visualised in Brat . .	155
Figure 4.15	The syntactic tree of a sentence containing an incorrectly tokenised abbreviation.	163
Figure 5.1	An example of citation, co-citation and citation coupling networks	171
Figure 5.2	A visualisation of the macro-level citation network extracted from the APh data	177
Figure 5.3	A visualisation of the meso-level citation network extracted from the APh data	179
Figure 5.4	A visualisation of the micro-level citation network extracted from the APh data	180
Figure 5.5	A screenshot of the interactive visualisation of the micro-level network extracted from the APh data . .	181
Figure 5.6	A screenshot of the interactive visualisation of the meso-level network extracted from the APh data . .	183
Figure 5.7	A screenshot of the interactive visualisation of the macro-level network extracted from the APh data . .	184
Figure 6.1	A screenshot of the secondary literature view in the Hellespont reading environment	194

LIST OF TABLES

Table 4.1	Mapping between named entities and relations, CTS identifiers and ontology classes.	125
Table 4.2	Number of documents and tokens contained in the L'Année Philologique (APh) dataset	127
Table 4.3	Basic statistics about the training set derived from the APh dataset	128
Table 4.4	Number and type of annotations contained in the APh training set.	129
Table 4.5	An example of the pipeline output	134
Table 4.6	Named entity extraction: selected examples of punctuation features.	138
Table 4.7	Named entity extraction: selected examples of case features.	138
Table 4.8	Named entity extraction: selected examples of number features.	138
Table 4.9	Named entity extraction: selected examples of pattern features.	139
Table 4.10	Named entity extraction: selected examples of semantic features.	140
Table 4.11	Evaluation results for the named entity extraction: overall precision, recall and F_1 score of CRF, MaxEnt and SVM.	154
Table 4.12	Evaluation results for the named entity extraction: precision, recall and F_1 score of CRF, MaxEnt and SVM, divided by named entity type.	155
Table 4.13	Evaluation results for the relation detection: precision, recall and F_1 score of the rule-based algorithm.	156
Table 4.14	Some examples of the relations that result from discursive citations.	156
Table 4.15	Evaluation results for the disambiguation of aauthor and awork entities and scope relations.	158
Table 4.16	Evaluation results for the disambiguation of aauthor and awork entities only.	159

Table 4.17	The 10 most ambiguous abbreviations of ancient works in the knowledge base	160
Table 5.1	Statistics concerning the APh data used to construct the three-level network	175

NOMENCLATURE

A&HCI	Arts and Humanities Citation Index 19, 20, 172
ACL	Association for Computational Linguistics 113, 117
ANTLR	ANother Tool for Language Recognition 147, 148
APA	American Philological Association 77
APh	L'Année Philologique 6–9, 76, 89, 90, 92, 99, 105, 121, 122, 125–130, 143, 150, 151, 164, 166, 168, 174, 175, 177, 179–185, 188
API	Application Programming Interface 5, 6, 79–84, 103, 104
BIBO	Bibliographic Ontology 75
BioNLP	Biomedical Natural Language Processing 117, 119
BiRO	Bibliographic Reference Ontology 75
BMCR	Bryn Mawr Classical Review 53, 54
Brat	Brat Rapid Annotation Tool 6, 7, 117, 122, 124, 131, 132, 135, 155
CFG	Context-free Grammar 147, 148
CIDOC	International Committee for Documentation 71
CIDOC CRM	CIDOC Conceptual Reference Model 5, 6, 62, 71, 72, 86, 88–91, 94, 96–100, 189
CiTO	Citation Typing Ontology 73–75
CLARIN	Common Language Resources and Technology Infrastructure 192
CoNLL	Conference on Natural Language Learning 113, 119, 152
CRF	Conditional Random Fields 152–154
CS	Computer Science 60, 110, 120

CTS	Canonical Text Services 5, 6, 8, 43, 44, 71, 78–85, 88, 96, 98, 99, 102–104, 124, 143, 150, 157, 166, 170, 176, 189
CWKB	Classical Works Knowledge Base 5, 6, 76–78, 84, 85, 101, 103
DARIAH	Digital Research Infrastructure for the Arts and Humanities 192
DEO	Discourse Elements Ontology 75
DH	Digital Humanities 15, 18, 19, 45, 170, 174
DoCO	Document Components Ontology 75
FaBiO	FRBR-aligned Bibliographic Ontology 74, 90
FRBR	Functional Requirements for Bibliographic Record 5, 68–72, 74, 79, 90, 95, 99
FRBR _{ER}	FRBR Entity-Relationship 71, 72
FRBR _{OO}	FRBR Object-Oriented 6, 62, 71, 72, 86, 88–91, 97, 99, 100, 125, 189
HTTP	HyperText Transfer Protocol 67, 68, 189, 192, 198
HuCit	Humanities Citation Ontology 6, 23, 49, 50, 52–54, 62, 63, 65, 68, 69, 76, 83, 85–101, 103, 104, 109, 189
ICOM	International Council of Museums 71
IE	Information Extraction 110, 116, 117
IFLA	International Federation of Library Associations and Institutions 69
IOB	Input, Outside, Beginning 7, 131, 132
JSON	JavaScript Object Notation 7, 149, 150
LAWD	Linked Ancient World Data 76, 83
LAWDI	Linked Ancient World Data Institute 83, 84
LIS	Library and Information Science 18, 120, 172
LOD	Linked Open Data 66–68, 76, 83, 84, 100, 101, 192, <i>Glossary: Linked Open Data</i>
MaxEnt	Maximum Entropy 152–154
MUC	Message Understanding Conference 119
NED	Named Entity Disambiguation 111

NER	Named Entity Recognition 21, 111, 113, 114, 119, 131, 152, 153, 165, 187, 188
NLP	Natural Language Processing 23, 63, 64, 110, 112, 113, 117, 119, 131, 132, 135, 147, 151, 162, 165
OCR	Optical Character Recognition 30, 43, 108, 126, 129, 130, 145
OED	Oxford English Dictionary 60
OWL	Web Ontology Language 65, 68, 88
PHI	Packard Humanities Institute 35, 42, 78
PoS	Part-of-speech 131, 134, 136, 137, 156, 157
RDF	Resource Description Framework 5, 6, 65–68, 76, 84, 90, 101–105, 189
SAWS	Sharing Ancient Wisdoms 44, 84, 88
SCI	Science Citation Index 19, 172
SPAR	Semantic Publishing and Referencing 74
SPARQL	SPARQL Protocol And RDF Query Language 65, 67
SQL	Structured Query Language 65
SVM	Support Vector Machines 152–154
SWAN	Semantic Web Applications in Neuromedicine 73–75
TEI	Text Encoding Initiative 43, 79, 82
TLG	Thesaurus Linguae Graecae 35, 38, 40, 43, 46, 70, 78
Turtle	Terse RDF Triple Language 6, 66, 85, 102, 105
URI	Uniform Resource Identifier 65–68, 84, 88, 189, 192, 198
URL	Uniform Resource Locator 77
URN	Uniform Resource Name 5, 6, 79, 80, 82–85, 88, 96, 98, 99, 102–104, 124, 143, 144, 150, 157, 166, 170, 176
W3C	World Wide Web Consortium 65, 101
XML	Extensible Markup Language 5, 6, 66, 81–83, 104

INTRODUCTION

Overview

This chapter introduces the focus of this research in section 1.1. The main research question that is being addressed is: how can a system to extract canonical references automatically from modern publications be developed? Section 1.2 outlines the motivations for this work and section 1.3 clarifies the nature of this research, specifically the role played by coding. In section 1.4 I introduce a key assumption, that canonical references constitute an essential entry point to bibliographic information for classicists. In section 1.5 I outline what is hoped to be achieved by this research. Finally, section 1.6 provides a guide for the reader to navigate through the contents of this dissertation.

1.1 THE FOCUS OF THIS THESIS: THE AUTOMATIC EXTRACTION OF CANONICAL REFERENCES

This research originally set out to address the following questions:

1. Is it possible to extract canonical references automatically from modern publications such as journal articles?
2. With what level of accuracy can this extraction be performed?
3. How can a system be implemented to perform this task?

As the research progressed, the importance of addressing the following additional question became clear:

1. How might the applications enabled by the automatic extraction of canonical references change the way we study classical texts?

Answering this last question is admittedly challenging as it requires an imaginative reflection on the transformative effects that tools yet to be built may have on scholarship. Yet, this reflection is an aspect that distinguishes digital humanities research from the mere development of digital tools (McCarty, 2014).

1.2 MOTIVATIONS FOR THIS WORK

What motivated this work is the sense of dissatisfaction I experienced with the tools for bibliographic search that are currently available to classicists. In particular, the dissatisfaction came from ascertaining that any search carried out by means of these tools is unlikely to be exhaustive given the sheer volume of bibliographic information that needs to be searched.

The difficulty in finding information relevant to one's research, which is common to many if not all disciplines, remains largely unspoken among classicists.¹ Yet, in Classics this problem is exacerbated by its fairly long history but also by the long shelf-life of publications as compared to other disciplines.

The tool I envisaged for my research would combine the granularity and specificity of an *index locorum* – an index of cited passages – with the ability to work on the larger scale that the increasing volume of available information requires. Indexes of cited passages are essential to classicists as they allow for precisely locating where a given text is cited within a publication. At the same time, the sheer volume of information renders the manual compilation of such indexes unfeasible, thus presenting us with the challenge of how to automate the extraction of canonical references.

This research does not focus on how such a tool can be implemented – although some of the functionality it could provide are sketched out in the last section of chapter 5. The focus, instead, is on the problem that

¹ Some exceptions, however, do exist: Calame (2001); Cozzo (2006); Cerri (2009). Interestingly, the words used in these studies to refer to this problem – “inflazione bibliografica” and “smarrimento bibliografico” – relate the phenomenon of (bibliographic) information overload to the effect of bewilderment resulting from it. Cozzo, in particular, takes this argument to a rather extreme point in his anthropological study of the tribe of classicists (2006, pp. 161-164). After observing that the choice of bibliography in this field is largely based on one's network of relations, he argues that the selection of bibliography is essentially governed by randomness, thus resulting in a weakening of the philological method of investigation.

needs to be solved before this tool can be built, namely how to capture canonical references automatically.

1.3 THE NATURE OF THIS RESEARCH

The research presented in this dissertation sits at the intersection between Classics and computing. This disciplinary area is also known as Digital Classics and falls within the broader category of Digital Humanities (DH) research. A good way to describe the nature of my research is to compare it with an exercise in translation. What is translated are, in this case, the mental processes and models underlying our ability to decode canonical references. What these processes and models are translated *into* is the formal language of computation. Ultimately, what is gained in the process – as often happens when translating a text from one language to another – is a better understanding of what has been translated.

Although not all scholarship in this area presupposes the ability to code or leads to the development of code as one of its outcomes, coding did play a primary role in this research. Code is the set of signs used to write the translation of the decoding of citations into a computationally solvable problem. In particular, since one of the questions that led to my research is *how can a system to extract canonical references be implemented*, the code that implements this system constitutes an essential part of the argument that unfolds throughout this dissertation.²

A key characteristic of this code is its development for research purposes, meaning that its goal is to demonstrate the feasibility of the approach I have developed rather than to provide the most efficient implementation. Since the code evolved along with the research, parts of it may be idiosyncratic or even obsolete. These parts may correspond to research directions that were explored and then abandoned. Nevertheless, every effort has been made to make the code and the data that were developed as part of this research openly available. In fact, this not only makes the code reusable by others but it also means that the approach described in this thesis is replicable.

² In turn, considering the code as scholarship requires us to develop strategies at the institutional level to assess and evaluate non traditional research outputs. This issue is currently being discussed in the DH community, see e.g. Presner (2012).

1.4 CITATIONS AND INFORMATION RETRIEVAL

The key assumption made throughout the thesis is that canonical references constitute an essential entry point to information for classicists. The special importance of these references is that they point directly to one of the main objects studied in classical scholarship – the ancient texts. This assumption is also justified by the needs and behaviours with regards to information retrieval that characterise humanities scholars in general, as has emerged from the literature reviewed in this section.

1.4.1 *Characteristics of References in the Classics*

Canonical references (or citations) are those references to ancient texts that are found in abundance within the secondary literature of Classics. Although the term *citation* may indicate also a direct quotation of a text, I use both *citation* and *reference* throughout this dissertation to mean an explicit act of reference in a written text. Moreover, I define primary and secondary literature as follows. Primary literature consists of the sources containing the evidence on which scholars base their scholarship, whilst secondary literature is made up of the publications in which scholars write up their scholarship.

Texts, however, are only one of the types of material that are typically cited. The range of materials is remarkably wide and includes inscriptions, papyri, manuscripts, coins and archaeological objects in general as well as other modern publications. Although references and citations can be found in any scholarly publication across the disciplines, references to primary sources in the Classics are particularly important as they refer to the very objects of the research.³

As a preliminary definition, canonical references are those that refer to texts in ways that are independent from any specific edition or translation of the cited text.⁴ Such references are valid no matter what edition one uses to look them up, thus the cited edition does not need to be specified. For example, given the citation “Hom. *Il.* I 1–10” – which identifies the first ten lines of the first book of Homer’s *Iliad* – the

³ The ubiquity of references led Unsworth (2000) to include *referring* – along with discovering, annotating, comparing, sampling, illustrating and representing – as one of the basic functions that are common to scholarly activity across the disciplines, what he calls “scholarly primitives”.

⁴ I discuss the nature of canonical references in more detail in section 3.2.

reader can easily look up this passage in the XIXth century translation into Italian by Monti or in the more recent critical edition by M.L. West.⁵

This research does not consider the extraction of references to those texts to which canonical ways of referring are not applicable. In fact, not *all* classical texts can be referred to in ways that are independent from a specific edition. Fragmentary texts, for instance, cannot be cited without using a specific edition as a reference. The reason for this is that editions of fragments may differ substantially from each other in terms of ordering and numbering, but also of attribution to a given author or work.

Another characteristic aspect of canonical references is their reliance on abbreviations. This can cause some challenges when trying to automatically capture canonical citations and their meaning. Indeed, canonical citations make up a complex notation system: learning how to use them and how to decipher them is part of the early training for anyone who wants to work in this field. Abbreviations within references can become so concise that in some cases they are impossible to decipher for non-classicists and can only be easily understood by “very few classicists” (Stephens, 2002, p. 67).⁶

Although these references are ubiquitous in publications in Classics, this study focusses on the extraction of canonical references from journal articles and bibliographic reviews. Indeed, another genre of publications where they play a prominent role are modern commentaries on classical texts. Although the extraction of such references from commentaries was not part of this research, the essential function performed by these references within commentaries is discussed later in this dissertation (section 2.1).

The references that were considered are exclusively explicit references to texts. Although not providing the explicit reference for a quotation in the context of a journal article is considered a sign of a flawed method-

⁵ However, in situations where the text as established by the editor in one specific edition is considered, one might want to add to the canonical reference the indication of a specific edition, e.g. “Hom. *Il.* I 1–10 (West)”.

⁶ Stephens argues that references in the Classics are “a sign system that aspires to scientific objectivity, but more often than not functions to exclude, as it were to prevent the uninitiated from penetrating the mysteries”. The example she provides to illustrate this effect is “Sch. in D.P. I, 317, 21 Bernh.”, which refers to an ancient commentary to a specific passage of Dionysius Periegeta’s *Orbis descriptio* according to the 1828 edition by G. Bernhardt. See also McCarty (2002, p. 381 n. 46) on how the style of referring is intentionally used by Dodds in his commentary to Euripides’ *Bacchae* as a subtle way to define his targeted audience.

ology, having implicit references may be totally acceptable in other situations, depending on the communicative context. In learned correspondence, for example, an implicit reference may be a deliberate communicative choice with the precise aim of stating the belonging of both the sender and the receiver to the same group.⁷

1.4.2 *Information Retrieval in the Humanities*

The decision to focus on canonical citations as a key entry point to bibliographic information was also based on reviewing the literature that investigated the needs and behaviours of humanities scholars with regards to finding information. The central aspects that emerged from this review are the centrality of references to primary sources and the usefulness of following chains of citations as a key strategy to information retrieval. In light of these findings, it is remarkable that the automatic extraction of canonical references to primary sources has not been previously considered as a means to improve the retrieval of bibliographic information.

Background

The needs and behaviours of humanities scholars has been an active area of study in the fields of Library and Information Science (LIS), DH and Bibliometrics. Indeed, all these disciplines have been concerned, to a varying degree, with building *something* for humanities scholars – be it a library system, a research infrastructure or a citation indexing tool.

Research on this topic started in the 1980s and early 1990s in the field of LIS with the work of Stone (1982), Ellis (1989), Watson-Boone (1994).⁸ Determining the information needs and behaviours of humanities scholars was essential for librarians in order to support scholars in their research by devising new library systems or by improving the guidelines for abstracting publications to cater for the specific needs of humanities scholars (Tibbo, 1993).

⁷ Examples of such implicit references can be found in the correspondence between the german philologist August Boeckh and Karl August Varnhagen von Ense Seifert (2014b,o). Implicit references such as “*χρυσέα χαλκείων*” (Hom. *Il.* 6.236) or “*procul negotiis*” (Hor. *Epod.* 2.1) imply the ability of the recipient of the letter to understand such references, hence his familiarity with the texts quoted.

⁸ For a thorough review of the early literature on this topic see Wiberley Jr. (2009, p. 2198) and Benardou et al. (2010, pp.19–21).

More recently, DH research has extensively built upon this study tradition as similar issues arose in relation to building new digital tools and research infrastructure for humanities researchers (Benardou et al., 2010; Blanke and Hedges, 2013; Benardou et al., 2013). The analysis of how research is conducted, with the aim of better understanding and supporting it, was the starting point for any attempt to build new tools or services and led to the development of general models of scholarly work and activities (Unsworth, 2000; Palmer et al., 2009; Anderson et al., 2010).

Finally, research in the field of Bibliometrics studied humanities scholars in order to gain a deeper understanding of their citation practices. Such an understanding is necessary to assess the suitability, usefulness and implications of the quantitative analysis of bibliographic citations found within humanities literature. Seminal work on this topic was carried out by Garfield (1980) and consisted of adapting the citation indexing system used for producing the Science Citation Index (SCI) to the characteristics of citations in the Humanities. His research was followed by more recent studies that discuss the utility of citation indexing systems for humanities scholarship in the light of the existing differences between citation practices of different disciplines (Hellqvist, 2010; Sula, 2012; Sula and Miller, 2014).

The Importance of References to Primary Sources

Humanities publications cite extensively primary sources. Research has found that roughly half of the citations contained within humanities publications refer to primary sources (Wiberley Jr., 2009, p. 2199). Such a tendency was already noted, albeit implicitly, by Garfield with regards to the Arts and Humanities Citation Index (A&HCI) (1980, p. 636). In fact, he notes that nearly 60% of the authors most cited within publications that appeared in 1977–78 were born before 1900 and 10% even lived before A.D. 140. Since in compiling these statistics he did not distinguish between references to primary and secondary sources, these figures can be explained in the light of the tendency to extensively cite the primary literature.

The importance of these references is also highlighted by Garfield's decision to include in the A&HCI references to primary sources such as texts, paintings or musical scores. Such a decision was taken notwithstanding the additional time and effort required for the index editors to

find and enter the various details into the database. The rationale for this decision was to enable users to formulate highly specific searches such as searching for publications that refer to Picasso's *Guernica* or *The Acrobat* and tracking the number of times these paintings are cited over a wider timespan (Garfield, 1980, p. 47). This way of searching consists of looking for publications that cite a specific primary source.

Garfield discusses also two specific issues raised by the indexing of this kind of references that are relevant in the context of this research. First, he reports that indexing the canonical references to the Bible challenged the data model of the index and required a work-around. This issue indicates the need for a dedicated data model to capture the structure of canonical references. The same issue is raised also by canonical references to classical texts and had to be addressed by the work discussed in this dissertation.

Second, Garfield observes that the cryptic abbreviations characterising some references in Classics publications proved to be hard to understand for the librarians working on the A&HCI, thus requiring a considerable amount of additional work. In fact, the librarians often had to read the broader context of a publication in order to understand which texts were cited in it. These two issues show that Garfield did already identify, as far back as the early 1980s, the main challenges raised by indexing canonical citations.

How do Humanities Scholars Look for Information?

What emerged from the literature reviewed in this section are also the key strategies for finding bibliographic information that characterises humanities scholarship. Firstly, scholars use proper names extensively when searching as compared with scholars in other disciplines (Wiberley and Jones, 1989; Bates, 1996; Palmer et al., 2009). A similar point is made by Crane et al. (2009). He argues that a digital infrastructure for research in Classics should facilitate way of searching for named entities that are especially relevant for classicists, such as authors, literary works and geographic places.

Secondly, a prominent behaviour among humanities scholars is to search for bibliographic information by browsing (Bates, 1989; Ellis, 1989; Meho and Tibbo, 2003). A typical example is browsing books in the stacks or shelves of a library. What characterises browsing as opposed to a targeted search is that it favours the serendipitous discov-

ery of relevant information: the physical proximity of books on library shelves, which is related to their subject classification, may in some cases transcend the boundaries of subjects.

Finally, a third prominent search strategy is the already mentioned *citation chaining* with its two variants of backward and forward chaining (Ellis, 1989; Buchanan et al., 2005). The former consists of starting from one publication – the seed document – and then following up the references it contains in order to expand the initial search and to discover other related publications. The latter consists of starting from a seed document and then finding which other publications cite it. Moreover, an empirical study of the information-seeking strategies of humanities scholars reports that searching and browsing proved to be rather ineffective strategies for locating information and that citation chaining was the most common behavioural pattern (Buchanan et al., 2005, pp. 227–228).

1.5 RESEARCH AIMS AND CONTRIBUTION

The problem that this research set out to address is the difficulty of carrying out exhaustive bibliographic searches in Classics. The approach I propose to deal with this problem consists of leveraging the references to classical texts contained within publications as an essential entry point to information. The ability to find publications citing a specific text passage of an ancient text is comparable to some extent to citation chaining.

Given how ubiquitous canonical references are within Classics publications, automating their extraction represents a key requirement for this approach in order to cope with the sheer volume of publications available. Garfield (1980), as discussed above, already understood the potential benefit of having such a system. Rydberg-Cox envisioned how in a digital library such a system could be used to group publications into clusters on the basis of the references to primary sources they contain (2006, pp. 65–67). More recently, Crane et al. (2009) argued that a digital infrastructure for research in Classics needs to enable scholars to search for canonical references and other named entities of interest. The approach I have adopted was informed by their study and consists of treating the extraction of these references as a problem of Named Entity Recognition (NER).

In order to make canonical references computable it was necessary to develop a formal model that allows us to represent the extracted references in a machine understandable format. This model also allows for publishing the extracted references in such a way that the published data can be reused in other contexts.

This research also suggests ways in which the ability to automatically capture these references may lead to new ways of studying classical texts. The main use of automatically extracted canonical references discussed relates to the search for bibliographic information. In particular, once such references have been captured, it becomes possible to allow scholars to search for publications that cite a specific set of text passages. Such a feature is likely to be of great use especially to those who are concerned with the study of specific textual matters (e.g. intertextual parallels).

Moreover, this research aims to lay the foundations for the quantitative analysis of the automatically extracted references. Once they have been extracted from text and represented in a digital format, this data lends itself to qualitative and quantitative analysis. It becomes possible, for example, to compute the frequency with which a given text passage is cited. If the citation data covers a wider temporal span it is possible to observe how this frequency varies over time, thus providing some insights into the diachronic variation of the number of publications that have discussed a given text passage.

1.6 READER'S GUIDE

The main audience of this dissertation comprises Classics scholars with a general interest in the application of computational methods to the study of classical texts. Therefore, every effort has been made to make this research as accessible as possible to a non-technical audience.

Chapters that discuss some highly technical matters – such as chapters 3 and 4 – are provided with a section where the key technical concepts are introduced: readers who are already familiar with these concepts may wish to skip those sections. Additionally, definitions of the key technical terms used in this dissertation are gathered in a glossary.

This dissertation is organised as follows:

- This introduction forms chapter 1 and presents the focus of this dissertation: the automatic extraction of canonical references from publications. This issue needs to be tackled to enable ways to search for publications citing a specific text passage, one possible solution to the problem of finding relevant bibliographic information in Classics.
- Chapter 2 focusses on one essential function that canonical references perform within commentaries, namely drawing the reader's attention to parallel passages. A diachronic sample of commentaries is examined in order to verify to what extent the practice of citing parallel passages evolved over time. Moreover, this chapter discusses the effects of the introduction of electronic concordances on the retrieval of parallel passages and provides an overview of the recent developments in the automatic detection of intertextual parallels.
- Chapter 3 presents the Humanities Citation Ontology (HuCit), a formal model of canonical references. This ontology formalises the conceptual model that our practices of citing texts already imply and allows for publishing in a machine understandable way the results of mining these references from publications. This chapter also examines the use of HuCit as the model for a database containing information that a computer programme needs to access in order to correctly interpret the extracted canonical references.
- Chapter 4 describes the approach I have developed to the automatic extraction of canonical references. This approach consists of adapting existing Natural Language Processing (NLP) methods to the extraction of this kind of reference. A scheme for the annotation of citations within texts is presented and an evaluation of the overall accuracy that can be achieved by using this approach is discussed.
- Chapter 5 explains how the implicit web of relations that canonical references constitute can be represented as a formal citation network. This network consists of three levels – macro, meso and micro – to allow for searching, visualising and analysing citation data at different levels of granularity. In particular, this chapter

discusses the use of this citation network for the purpose of searching through large-scale sets of publications.

- Chapter 6 concludes this dissertation by discussing the contribution it has made to Digital Classics research and by outlining the new research perspectives that it potentially opens up. This chapter discusses how the system for the automatic extraction of canonical references could be further improved, extended and made available to the wider community. Moreover, other potential areas for the application of the work presented are discussed.

CANONICAL REFERENCES AND PARALLEL PASSAGES: BETWEEN TRADITION AND INNOVATION

Overview

This chapter focusses on one specific function that canonical references perform within modern commentaries, namely drawing the reader's attention to parallel passages. In section 2.1 I examine a diachronic sample of classical commentaries in order to determine to what extent the practice of citing parallel passages evolved over time. In section 2.2 I discuss the effects that the introduction of electronic concordances have had on the *modus operandi* of classicists with a specific focus on the retrieval and discovery of intertextual parallels. Finally in section 2.3 I put my research into the broader context of the current developments in Digital Classics research.

2.1 CITING PARALLEL PASSAGES: A DIACHRONIC PERSPECTIVE

Although canonical references are ubiquitous within publications in the Classics, in the context of modern commentaries they perform the specific function of indicating parallel passages. Such passages – which may be drawn from other works by the same author as well as from works by different authors – are cited by the commentator to help contextualise or elucidate a given passage of the commented text.¹

Finding secondary literature that discusses a given set of parallels (e.g. journal articles) is one of the activities where the automatic indexing of cited passages could prove most useful to classicists. Therefore, the

¹ For a more precise classification of the various functions of parallel passages see Gibson (2002), discussed *infra* at p. 33.

function of canonical references to indicate parallel passages deserves particular attention in the context of this research.

In this section I consider a selection of commentaries on classical texts and look at how the practice and rhetorical function of citing parallel passages has evolved over time. Such an analysis is necessary for two reasons. First, it is important to take into account possible diachronic variations of the structure of canonical references. These variations need to be taken into account while designing a system to extract canonical references and a scheme to annotate them within the text. Second, since the rhetorical function of a canonical reference is not captured automatically, an awareness of how this function changed over time constitutes the theoretical framework necessary for a full understanding of the results of the automatic extraction.

2.1.1 *Selected Examples of Classical Commentaries*

The commentaries I have selected are all of considerable importance in the history of classical scholarship and were mostly published in the period between the end of the XIXth and first half of XXth century, which is considered the acmé of the modern learned commentary.² The decision to include also the XVIIth century commentary on the *Aeneid* by La Cerda deserves some explanation.

The identification of a precise date for the emergence of citations comparable with contemporary canonical references is problematic as the practice of citing other texts, albeit more or less explicitly and accurately, has been attested since antiquity.³ Nevertheless, the XVIIth century is a reasonable *terminus post quem* since the need to provide enough details in a citation to allow the reader to precisely locate the cited passage started to emerge during this period. Hauptman, in his history of documentation from the antiquity to the present, cites various example citations that confirm the emergence of such awareness among book editors around the first half of the XVIIth century.⁴ Around the same time

² See Grafton et al. (2010, pp. 225–233).

³ On this topic see e.g. Higbie (2010), in particular pp. 2–14.

⁴ Among the examples that Hauptman provides, the example drawn from an English translation of Boethius' *Consolatio Philosophiae* (1609) is of particular interest in this context. The translator decided to add the marginal reference "Ovid. lib 2. Metamor. E. Macrobius. Lib 1. Saturna 1" to elucidate an oblique citation contained in the original text "Thou has heard in the Poet's Fables how the Gyants..." (2008, pp. 22–24).

the need emerged for long sequences of text to be split into smaller chunks. In the first instance the splitting facilitated a more enjoyable reading experience and an easier memorisation of the text, but later it also allowed the reader to cite texts in a more accurate way. Examples of this practice can be observed in the activity of notable humanist printers like Stephanus.⁵

2.1.2 XVIIth century

The first example to be considered here, the commentary on the *Aeneid* by de la Cerda (1612), comes from around the same period and shares some similarities with the examples given by Hauptman. Despite being dated this commentary is considered modern for its period and is still used and cited in recent studies on Vergil.⁶

The references contained in La Cerda's commentary are characterised by a different and slightly inconsistent use of punctuation compared to what we are used to nowadays. At the same time, they are not dissimilar from contemporary references as they contain most of the information that is necessary to precisely locate the cited text.

La Cerda's way of referencing is noteworthy also for the following reasons. Firstly, while names of authors and titles of works are most often abbreviated, citations still feature overall a very discursive form as a result of their harmonisation with the grammatical and syntactical context of the sentence. This characteristic can be observed in references such as "scribente sic Platone lib. II. Leg. [...]" or "Ammian. lib. 17" (see figure 2.1 a-b).

Secondly, the structure of citations in La Cerda tends to be less consistent or standardised than it currently is. The main difference is the order in which the different components of the citation – author, work and cited passage – are given. The commentator for example writes "Ouid. 4. Trist. Eleg. I." and "Horatio Od. 11 lib. 1." whereas nowadays one would more commonly write "Ovid. *Trist.* 1.4" and "Hor. *Od.* 1.11" (figure 2.1 c-d).

Thirdly, compared to the current standards of citing ancient sources, his citations appear considerably less granular. For instance, the reference to Silius' *Punica* book 8, line 405 in the commentary to *Aen.* 7.49, is

⁵ On the importance of Stephanus' edition for the way of citing Plato see *infra* p. 57.

⁶ For an in-depth study on this commentary see Laird (2002).

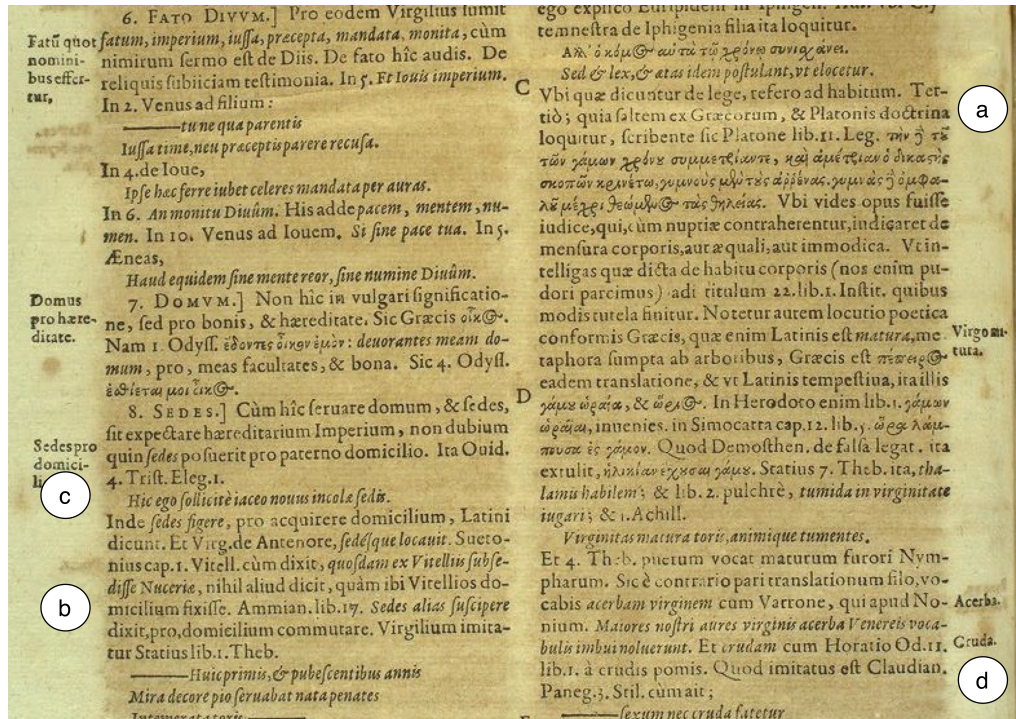


Figure 2.1: La Cerda's commentary on the *Aeneid* (1612): examples of canonical references to Plato (a), Ammianus (b), Ovid (c) and Horace (d).

given as "*Silium lib. 8 et tullo sanguis ab alto*", with the literal quotation to obviate the omission of the line number.

2.1.3 XIXth century

Lachmann on De Rerum Natura

The second example to be considered here is the commentary to Lucretius' *De Rerum Natura* published by Lachmann (1850) some 200 years later.⁷ This commentary derives from the critical edition of Lucretius' text, published in the same year, in which Lachmann demonstrates his method of reconstructing the relationships between the various manuscripts of a text as a tree or *stemma*, having at its origin the manuscript from which all others descended, the archetype.

The works of La Cerda and Lachmann are both written in Latin and feature an almost identical presentation of the *lemmata*. References to primary sources in Lachmann, however, differ substantially as they are characterised by a more consistent structure and a greater precision

⁷ The first edition of the commentary to books 1–6 appeared in 1612; the commentary to books 7–12 was published in 1617 and is reproduced here in its 1647 reprint.

50

I

enim Iuvenalis xiii,98. quamquam video hos ablativos ab aliis corripī. Horatius sermonum ii,3,193 *heros ab Achille secundus*. in Herus epistula (non scripsit eam Ovidius) 148 *A tibi suspecto ducit Ulixē genus*. Statius Thebaidos xii,348 *parvoque torum Polynice fovebo*. ita Propertius in v,11,40 putatur scripsisse *fregit Achille domos*. certum est in *Diomede Ganymede Lycomede* ultimam semper corripī: neque id mirum, cum Varro vel pluralem Latine extulerit *Diomedes* et *Diomedibus*, de lingua Latina x p.573, neque eum ita inflectendum censuerit ut dixerunt Graecorum more Cicero de oratore ii,94 et Gellius xiv,1 et 6 *Naucratae Socratae Antisthenae Hippocratae*, Iuvenalis ii,7 *Cleanthas*, *Hermeraclas* Cicero ad Atticum i,10,3. quo magis credibile est Vergilium in Aeneidos xi,243 scripsisse *Diomedem Argivaeque castra*, ut Statius scripsit in Achilleidos ii,217 *Lycomedem adfatur in armis*. Terentianus Maurus 2059 dixit *Diomedem*, quo poetas veteres usos esse non inveni.

741. MAGNO CECIDERE IBI CAUSA. Corrector oblongi CASU.

Figure 2.2: Lachmann's commentary on Lucretius' *De Rerum Natura* (1850): an excerpt of the commentary containing several canonical references.

when indicating the cited passage. Figure 2.2 shows some citation examples such as "Horatius sermonum II,3,193", "Statius Thebaidos XII,348" and "ita Propertius in V,11,40". Moreover, these references do not yet display the high density of abbreviations that has been characterising canonical references since the end of the XIXth century.

Two Commentaries on Tragedies

The third set of examples to be considered consists of two commentaries published towards the end of XIXth century: the commentary on Euripides' *Herakles* by Wilamowitz (1895) and the commentary on Sophocles' *Electra* by Jebb (1894).

Both commentaries are of great importance: Wilamowitz's is considered the first commentary on a Greek tragedy and a fundamental book

in classical scholarship⁸, while Jebb's is considered to have set the standard for judging editors of Sophocles.⁹

The vernacular languages in which both are written – German and English respectively – constitute already an innovative aspect as opposed to the Latin of Lachmann's commentary, published some 50 years earlier.

However, it is in the style of citing ancient texts that the distance from Lachmann is more evident. Both commentaries display a compact citation style, which relies heavily on the abbreviation – and sometimes even the omission – of the author and title of the cited work. In the case of Jebb's commentary such a compact style is also justified by the choice of layout. The commentary is arranged in two columns and printed underneath the established text, from which it is separated by the critical apparatus (see figure 2.3).¹⁰

The differences between these commentaries with regards to the citations of parallel passages are minimal. The citations in Jebb's commentary use the Latin abbreviations, whereas Wilamowitz abbreviates the German names and titles (e.g. "Thuk." for Thukydides). A further stylistic difference observed is Wilamowitz's preference for the Alexandrian way of referring to the Homeric poems.¹¹ An example of this style is the reference "Hom. Φ 270" shown in figure 2.4.

By the end of the XIXth century the canonical references that introduce parallel passages reached the standardised form that they have exhibited ever since. The differences observed in later commentary examples consist merely of small variations within a structure and a format that had by then been largely standardised.

2.1.4 XXth century

The next two examples are drawn from Pease's commentary to the *Aeneid* (1935) and Fraenkel's commentary to Aeschylus' *Agamemnon*

⁸ Briggs and Calder 1990, p. 498.

⁹ See Briggs and Calder (1990, p. 242). They also note that his contemporary G. Kaibel plainly ignored Jebb's work in his commentary to the same tragedy published a few years later.

¹⁰ It is worth noting that Jebb's commentary is an apt example of a sophisticated layout that can make the Optical Character Recognition of commentaries quite challenging, particularly the layout recognition. In fact, correcting the OCR errors caused by the elaborate layout usually requires a considerable amount of post-processing to prepare the output for text mining and information extraction.

¹¹ On this particular citation system see *infra* p. 56.

Πυλάδῃ, τί χρὴ δρᾶν, ἐν τάχει βουλευτέον·
 ὥς ἡμῖν ἤδη λαμπρὸν ἡλίου σέλας
 ἔῶα κινεῖ φθέγματ' ὀρνίθων σαφῇ,
 μέλαινά τ' ἄστρον ἐκλέλοιπεν εὐφρόνη.
 πρὶν οὖν τιν' ἀνδρῶν ἐξοδοιοποιεῖν στέγης, 20
 ξυνάπτετον λόγοισιν· ὥς ἐνταῦθ' ἔμην

τόδε, | τῆς οἰστρ. ἄλσος κ.τ.λ. 10 τε] δὲ T. 11 φόνων] φονῶν Dindorf.
 12 κάξεθρεψάμην] καὶ σ' ἐθρεψάμην schol. Hom. *Il.* 2. 485. Steinacker conj. κάν-
 εθρεψάμην. 14 τιμωρὸν φόνου made from τιμωρῶν φθόνου in L. 15 This verse
 was omitted in the text of L, and added in marg. by the 1st hand. Nauck brackets
 the words Ὁρέστα...Πυλάδῃ, thinking that Pylades had no place in the genuine play.

9 φάσκειν (infin. as imperat.), = 'deem,'
 'believe': *O. T.* 462 n.

Μυκήνας. This plural form (the pre-
 valent one) occurs in *Il.* 2. 569, 4. 376;
 but elsewhere metrical convenience led the
 Homeric poet to prefer the sing. *Μυκήνη*,
 which allowed him to prefix *εὐρύγυνια* (*Il.*
 4. 52), and *πολυχρύσοιο* (*Il.* 7. 180, 11. 46;
Od. 3. 305).

which he had brought from Asia to a poor
 country. Helbig (*Das hom. Epos aus den*
Denkm. erläutert, p. 50) thinks it certain
 that the precious metals became scarcer
 in the Peloponnesus after the Dorian con-
 quest. When the Spartans, in the first
 half of the sixth century, required gold
 for a statue of Apollo, they had to procure
 it from Sardis (*Her.* 1. 69).

Figure 2.3: Jebb's commentary on Sophocles' *Electra* (1894): an excerpt showing the layout of the commentary, which has the text and critical apparatus at the top and at the bottom the commentary arranged in two columns.

(1950). These examples show to what extent the same intertextual parallel can receive a substantially different treatment by different commentators. The parallel at issue is the one between the Vergilian verse "exoriare aliquis nostris ex ossibus ultor" (*Aen.* 4.625) and the Aeschylean verse "ἤξει γὰρ ἡμῶν ἄλλος αὖ τιμάορος" (*Agam.* 1280).

Pease's commentary to *Aen.* 4.625, on the one hand, discusses a number of parallels related to the verb "exoriare" and concludes with a cursory remark on the resemblance with a verse in Aeschylus: "[w]ith the line in general cf. Aesch. *Agam.* 1280: ἤξει γὰρ ἡμῶν ἄλλος αὖ τιμάορος, κτλ.,". The similarity is noted and brought to the reader's attention but the commentator does not further characterise the intertextual parallel. It is worth noting that Pease's commentary – a book of some 500 pages covering only book 4 of the *Aeneid* – is often cited as an example of commentaries that are valuable more for the sheer number of parallel passages they contain rather than for their critical remarks.¹²

Fraenkel, on the other hand, in his lengthy commentary to the *Agamemnon* (over 800 pages) provides an articulate discussion of the similarity between the two verses (see figure 2.5). He argues that Vergil is con-

¹² See Fowler (1997, p. 14) and Fowler (1999, p. 436).

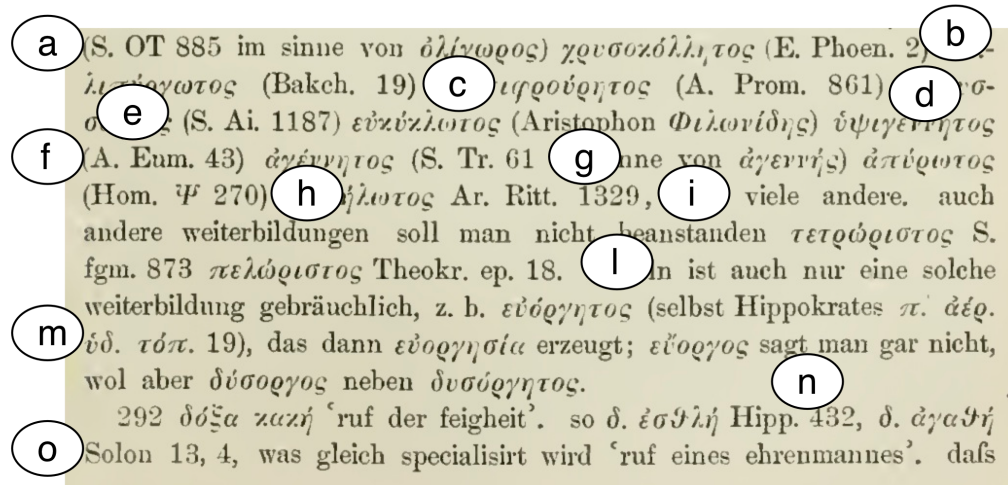


Figure 2.4: Wilamowitz commentary on Euripides' *Herakles* (1895): an excerpt of the commentary containing several canonical references (a-o); (h) an Alexandrian reference to the *Iliad*.

sciously borrowing from Aeschylus in light of the overall tragic tone of the episode of Dido and additionally cites secondary literature in support of his argument.

2.1.5 XXIst century

The passage of Pease's commentary discussed in the previous section is an early example of a use of parallel passages that becomes more radical in XXIst century commentaries. This use, which is a much criticised aspect of some contemporary commentaries, consists of providing parallel passages with little or no further characterisation of their relevance to the commented text and often arranged in relatively long lists.

In an article entitled "Cf. e.g.: a typology of 'parallels' and the function of commentaries on Latin poetry", Gibson provides an apt summary of the main point of discussion when he asks:

[w]hat critical processes are elided in the abbreviation – cf. e.g. – which routinely introduces parallels in commentaries?" (2002, p. 331).¹³

One criticised tendency is the use of formulaic expressions to introduce such parallels, first and foremost "cf e.g.", the "infamous opening" as Fowler calls it (1999, p. 434). In some cases these formulaic

¹³ His observation is echoed by Goldhill: "[w]hen we are asked to compare, what is being compared and how?" (1999, p. 397).

line 1279

COMMENTARY

For οὐ μὴν . . . γε Sidgwick quotes S. *Oed. R.* 810 οὐ μὴν ἴσῃν γ' ἔτεισαν; for further examples, see Denniston, *Particles*, 335.

1280. ἄλλος: 'alius ac nos, qui mortui iacebimus' (Klausen). Schneidewin and others have referred this to the so-called pleonastic use of ἄλλος, but it is not quite of the type to which this term is usually applied, since τιμάορος must be taken predicatively with ἦξει; cf. on 530. For ἄλλος αὖ cf. Ar. *Thesm.* 664 εἴ τις ἐν τόποις ἐδραῖος ἄλλος αὖ λέληθεν ὧν and ἕτερος αὖ, common in Aristophanes.

τιμάορος: cf. on 514.

It seems to me certain that the famous line (Virg. *Aen.* 4. 625) *exoriare aliquis nostris ex ossibus ultor* was consciously borrowed from 1280.¹ Even the sequence of the words has been to a great extent preserved. The loving care with which Virgil chooses ornaments from Greek tragedy to adorn the last speeches of Dido is characteristic of his admiration for the great poetry of Athens. This is not a matter of mere details. It has long been realized that 'the episode of Dido is worked out very much in the spirit of the Greek tragedy' (H. Nettleship, *Suggestions to a Study of the Aeneid*, 1875, 34).

Figure 2.5: Fraenkel's commentary on Aeschylus' *Agamemnon* (1950): the commentary to line 1280 where Fraenkel discusses the intertextual relation between this line and Verg. *Aen.* 4.625.

expressions become so elliptic that it is almost impossible to see the interpretative value of the parallel or to reconstruct the underlying critical thinking of the commentator.¹⁴

687–98 The *locus amoenus* is a standard setting for violence and rape; cf. e.g. *Hymn. Dem.* 6ff. (Persephone), Callim. *Lau. Pall.* 71ff. (Teiresias), Moschus *Eur.* 63ff. (Europa), Prop. 1.20.33ff. (Hylas). In the *Metamorphoses* a pool, set in wooded and shady surroundings, is used recurrently as a scene for violence, death, and rape; cf. e.g. *Met.* 2.417ff. (Callisto and Jupiter), 2.454ff. (Diana and Callisto), 3.155ff. (Actaeon), 3.407ff. (Narcissus), 4.296ff. (Hermaphrodite), 5.385ff. (Persephone), 5.585ff. (Arethusa), 10.126ff. (Cyparissus); also *Fast.* 3.13ff. (Silvia and Mars), Parry *TAPA* 95 (1964) 275–80, Segal, *Landscape in Ovid's Metamorphoses* 4–19.

Figure 2.6: Gibson's commentary on book 3 of Ovid's *Ars Amatoria* (2003): a long list of parallels related to the convention of using the *locus amoenus* as a setting for violence (*Ars.* 3.687–98).

Another criticised tendency is to provide long lists of parallels that are of little value to improve the interpretation of the commented text.

¹⁴ Gibson defines a typology that classifies parallel passages based on the following functions: 1. establishing the text; 2. comprehending the text; 3. establishing register within the text; 4. contextualising the text; 5. identifying intertexts/allusions; 6. identifying *topoi*; 7. 'supplementing' the text (2002, pp. 333–346).

Gibson gives an example of this second tendency drawn from his own commentary to Ovid's *Ars Amatoria*. In the commentary to lines 687–698 he cites a long list of parallel passages on the motive of the *locus amoenus* as a setting for violence (figure 2.6). The effect of this list, he notes, is to bury the text in conventions rather than helping the reader better to understand the meaning and effect of setting violent acts in a *locus amoenus* (2002, pp. 351–352).

A few contributions to this debate on the classical commentary seem to relate these tendencies, more or less explicitly, to the availability of digital tools, especially electronic corpora and concordances. Although more research is needed to verify this assumption, there is little doubt that these tools have altered scholarly practices in the Classics including the preparation of commentaries. The introduction to Heyworth's *Cynthia* (2007), his commentary on the text of Propertius, contains an interesting reflection on what it meant for his research to have access to the entire Classical Latin literature in a searchable format (emphasis my own):

Suddenly it was possible to answer questions that one had hesitated to formulate because the time required to gather the evidence made the effort impracticable. On the other hand, when such tools are widely available, there is less value in lists of usages of particular words, **and I hope that I have not too often fallen into the trap of giving such information simply because I was able to do so** (2007, p. ix).

To summarise, the appearance and structure of references to parallel passages evolve diachronically following perhaps the gradual professionalisation of the Classics as a discipline. First, canonical citations move from a clearly discursive form to a more structured and formal one. This form is characterised by the abundant use of abbreviations, which seems to mimic the formal notation systems that are used in the sciences. Second, they become more granular as the cited passage is specified in a more and more precise way (e.g. line numbers are given in addition to book numbers). Finally, the rhetorical function of citing parallel passages undergoes a gradual radicalisation. In its more extreme manifestations such a radicalisation leads to the sole use of expressions like “cf e.g.” to introduce citations as well as to the habit of constructing relatively long lists of them.

2.2 RETRIEVING PARALLEL PASSAGES: ELECTRONIC CONCORDANCES

In this section I attempt to discern some of the changes that the introduction of digital tools has had on the way classicists work. I focus in particular on the use of electronic corpora – e.g. Thesaurus Linguae Graecae (TLG), Packard Humanities Institute (PHI), etc. – and electronic concordances – e.g. Ibycus, Diogenes, Musaios, etc. – for the retrieval of parallel passages.¹⁵ This is, in fact, an activity that is going to be affected by the tools that the automatic extraction of canonical references enables. Moreover, looking at the effects of these resources helps us imagine the effects that a citation extraction system may have on the work of classicists.

2.2.1 *Classical Scholarship and Computer Technology*

Many studies have focussed on the relation between classical scholarship and computer technology.¹⁶ The term *impact* is often used in these studies to describe such a relation. However, this term may seem reductive as it implies that the process of technological change has already been completed and we are therefore in the position of observing its effects (Connor, 1990). It may still take some time until computing technology realises its full potential, as Connor rightly observed. Nevertheless, it is certainly already possible to discover scholars' reflections on the differences between traditional and digital tools and on how the latter influence research questions and alter existing research practices.

The impression one may have when trying to find such reflections within classical scholarship is that classicists have simply neglected to consider this relation (Ruhleder 1995, p. 41; Connor 1990, p. 59). Closer investigation of the problem, however, reveals that such written accounts do exist but are in fact scattered and so hard to find. The account given in this section is hardly exhaustive as finding traces of these reflections by sifting through archives such as JSTOR, Google Books or the Internet Archive proves challenging in two respects. First, classicists do

¹⁵ This section, however, does not aim to give a comprehensive account of the studies that focussed specifically on the TLG or the PHI. A good starting point on this respect is Babeu (2011, pp. 1–7).

¹⁶ See e.g. Ireland (1976); Connor (1990); Crane et al. (1991); Bolter (1991); Ruhleder (1995); Hardwick (2000). See also the series of papers on "Classics and the Computer" (McDonough, 1967; Waite, 1970; Brunner, 1993; Crane, 2004).

not cite systematically the digital tools used in their research. Second, a great manual effort is needed to filter out those publications that, despite mentioning explicitly these tools, do not reflect on their influence on scholarship.¹⁷

Citing the Tools Used

Indeed, classicists' attitudes towards citing electronic resources and tools varies considerably. Some scholars do cite the tools they use, occasionally meticulously, as is the case with Richardson with regards to his study of the idea of Roman imperialism (2008). Not only does he acknowledge in the preface the tools employed to find all occurrences of the words *imperium* and *provincia* within Latin literature, but he also discusses further technical aspects of his method in a separate article (2005). In contrast, many other scholars do not mention such tools explicitly as this is something they take for granted, and for the same reason they do not mention every single concordance or index they checked.

Citing the tools used is as essential to a rigorous and sound method as being accurate in the citation of sources. Scholars who make use of electronic corpora for their research and want to follow a rigorous method also with respect to the use of these corpora ought to provide not only name and version of the electronic corpus and the software used to access it, but also some indication of how the search queries were constructed. Two aspects of digital tools make such a practice necessary.

First, the changes introduced by electronic corpora affect not only the form under which texts are presented and represented but also the content itself (Ruhleder, 1995). Editorial decisions, which in a printed edition or concordance are normally explained by means of footnotes, are embedded in the computational artifact with the resulting risk of becoming invisible. Such decisions concern for example which texts were included in the corpus, which edition of a text was chosen and how texts were processed and transformed (for example due to routine checks for correction purposes).

Second, changes in the corpus, or in the software used to interrogate it, may cause the same search to return different results when performed

¹⁷ I would like to thank the members of the Liverpool Classicist discussion list that have drawn my attention to many of the essays discussed in this section. My original posting to the list can be found at <http://listserv.liv.ac.uk/cgi-bin/wa?A2=ind1405&L=classicists&T=0&P=17135>.

at different points in time. These changes may be related to the search algorithms or the format used to encode and store the data. The matter is further complicated when the search is conducted against or by means of online resources. Indeed, many of these resources are not versioned and, even when they are, changes in these resources – especially if small – may not always coincide with a new version.¹⁸

2.2.2 *Tools Shape Research Questions*

The first aspect of the relation between scholarship and technology to consider is how tools influence research activities. Not only do they define the array of research questions that we are able to ask by using them, but they also determine to some degree what questions we decide to pursue (Crane et al., 1991, p. 293). While electronic corpora and concordances certainly widen the array of hypotheses that can be explored, their characteristics and capabilities inevitably impose some limitations on which questions they allow us to investigate (Connor, 1990).

Increased Speed

The increased speed of operations is arguably the most distinctive trait of electronic concordances. Indeed, it takes now a relatively short amount of time to check all the occurrences of a given word by means of an electronic concordance whereas it would have meant a “lifetime’s project for a sizeable team of scholars” even fifty years ago (Richardson, 2005, p. 139).

There are questions that simply require such a high speed of processing to be answered, for example the question that led Fr. Roberto Busa to create automatic concordancing. He created the *Index Thomisticus*, considered to be the first electronic concordance, to be able to check systematically the occurrence of words within the corpus of Thomas Aquinas. He realised that checking all the occurrences of not only content words but also of function words, especially the preposition “in”, was necessary to understand Thomas Aquinas’ concept of divine immanence. Such a task could have not been done manually, but it was feasible with the use of computation.¹⁹

¹⁸ Sosin (2014) discusses an interesting example of this problem concerning the Duke Databank of Documentary Papyri.

¹⁹ On the significance and history of the *Index Thomisticus* see Winter (1999).

Speeding up the work, however, should not be the primary goal of using computation for humanities research, as Busa himself repeatedly wrote. In an article where he reflects about the 30 years of history of the *Index Thomisticus* Busa notes:

[...] the use of computers in the humanities has as its principal aim the enhancement of the quality, depth and extension of research and not merely the lessening of human effort and time (1980, p. 89).²⁰

Indeed, electronic corpora and concordances did improve the extension of research insofar as they allowed scholars to verify a higher number of hypothesis before deciding to pursue a given research question. This change is well reflected in one of the interviews reported by Ruhleder in which a scholar reflects on what changed with the advent of the TLG. The scholar recalls how, before the introduction of automatic search, “[you] can only play out a certain number of your hunches, so you go with the best ones” (Ruhleder, 1995, p. 48). In other words, electronic concordances influence the direction that the research takes: they free us from the need of being selective with regards to the number of occurrences that *can* be checked.

Systematic Verification

The ease with which an electronic concordance allows us to check the occurrences of a word can have other quite different effects. On the one hand, it may lead to the use of the tool for its own sake. Indeed, in the introduction to his book quoted above, Heyworth describes the use of such a tool to produce long lists of parallel passages as a trap he tried to avoid while preparing his commentary.

On the other hand, this ease and speed of operation enables the verification of hypotheses that were formulated empirically. An example of this verification is given by Cowan (2013) in an article where he discusses a fragment of Sallustius’ *Empedoclea* contained in a letter by Cicero. The use of electronic concordances and corpora of Latin literature allowed him to verify, and thus to accept, a judgement concerning the juxtaposition between *vir* and *homo* that was already formulated by Housman in the early XXth century.²¹ This ability to verify other schol-

²⁰ On the perception of computing as a means of saving labour and time see also McCarty (2013, pp. 4–5).

²¹ See Cowan (2013) pp. 764–765 and n. 5.

ars' hypothesis represents a considerable novelty for disciplines such as literary studies. Indeed, scholars' theses and arguments in these disciplines can seldom be verified; the main criterion for acceptance is their ability to persuade the reader.²²

Similarly, this systematic verification is one of the foreseeable uses of automatically created indexes of cited passages. Once scholars have at hand an index of the passages cited within the journal articles contained in an archive such as JSTOR, they can verify whether a given set of parallels has been previously discussed in almost any already published article. The bibliographic searches that the existing tools allow for, instead, are far from being systematic or even just exhaustive. Indeed, while well educated scholars can be confident that they have found the most important articles about a parallel of interest, we can hardly be sure to have found *all* articles addressing it.

Searching for the Tangible

A further aspect of digital tools that needs consideration is how their characteristics impose some limitations on the range of questions that these tools enable us to address. In the specific case of electronic concordances, what they allow us to accomplish is to search for tangible *things* that are explicitly mentioned in the text. As a result, they are suitable to find out where and how often a given word or personal name is referenced but not to study what is not literally mentioned in a text.

Let us consider now some examples of research questions that challenge the current abilities of electronic concordances. A first example is provided by Gioseffi in his investigation of the use of the formula "id est" within the late antique commentaries to Vergil (2008). He argues that a study of this formula cannot rely exclusively on searching for occurrences of "id est" within a corpus of Latin texts. In fact, such a search would inevitably fail to capture other expressions such as parenthetical sentences that do not contain these words but do perform a similar explicative function.

A second example of problems that electronic concordances are ill-suited to address are "imported concepts" (Connor, 1990, p. 60). These are modern concepts that, despite not having a correspondent term in

²² Jockers argues that the large-scale, quantitative analysis of literary texts brings into the Humanities some degree of verifiability and repeatability that characterise experimentation-driven research in the Sciences (2013, pp. 5-10).

the ancient language, can be applied to study ancient civilisations (e.g. ideology, imperialism or neutrality).

A final example is the study of intertextual parallels. Electronic concordances are useful in cases where an allusive effect is obtained by using an almost exact repetition of words that appear in another text. Conversely, they cannot help us identify those intertextual parallels that consist of a thematic similarity or that are triggered by the use of a partly different wording.²³

2.2.3 *Reading, Remembering, Retrieving*

The second aspect of the relation between scholarship and technology to consider is how tools may alter scholarly practices. In particular, electronic concordances intervene on two key aspects of the retrieval of parallel passages, namely reading and remembering.

Reading and Retrieving

Close reading – i.e. the “sustained, concentrated reading of text” (Jockers, 2013, p. 6) – has been to date the main way of familiarising oneself with any body of texts and the primary practice in literary studies.²⁴ Reading helps us internalise the texts by organising them into a network of associations between what we are reading and what we have already read. These associations determine, in turn, our ability to retrieve – i.e. bring back to memory – passages of the texts read. If we accept this model as an approximation of how memory works, it seems fair to assume that the way in which we read influences our ability to remember what we have read.²⁵

One criticism that has accompanied the adoption of technologies like the TLG in the Classics is that they risk eliminating the need to read the texts, thus exempting scholars and students alike from having to read the texts in order to be able to do research (Ruhleder, 1995, p. 49). Despite this criticism, electronic concordances certainly did not render

²³ See section 2.3 for a detailed discussion of methods and tools that have attempted to overcome this specific limit of electronic concordances.

²⁴ This kind of reading is referred to as *close reading* in order to distinguish it from the large-scale, quantitative analysis of texts, the so-called “distant reading” (Moretti, 2007) or “macroanalysis” (Jockers, 2013).

²⁵ Yet, investigating *how* the use of external databases and automatic search functionalities affect reading and memorising falls outside the scope of this work. Some stimulating reflections on this topic are contained in Barnett (2013), especially chapter 6.

the close reading of texts unnecessary. What they did, however, was to provide scholars with an additional means of retrieving text passages that is fundamentally different from the close, linear reading of text.

Although printed concordances and indexes arguably already provide nonlinear ways of accessing texts, such a nonlinearity has been radicalised by the introduction of electronic corpora and concordances.²⁶ In fact, they enable us to search quickly entire corpora as if we were consulting several printed concordances simultaneously. Moreover, the possibility of using boolean operators and regular expressions within a search allows for more flexible and sophisticated searches than it was possible to carry out by using printed concordances. All these characteristics favour nonlinear ways of accessing texts and make electronic concordances ideally suited to supplement our ability to remember.

Remembering and Retrieving

In addition to altering the linearity of reading and digesting texts, electronic concordances amplify the serendipitous nature of how memory works. In fact, the results returned upon a search often contain matching passages that the memory did not suggest. This aspect of amplified serendipity is essential in an area such as the study of intertextuality where finding similarities and correspondences between texts is the very essence of the research.

A remarkable reflection on this interplay between computer searching and human memory in relation to the retrieval of parallels is provided by Fowler (1997). In this article, which discusses theoretical issues related to intertextuality, Fowler also takes issue with the complaint that much of the work on intertextuality is based not on the thorough reading of texts but on computer searching (1997, p. 19). He replies to this criticism by providing an example, drawn from his own experience, highlighting how computer searching can be seen as an enriching complement to human memory rather than a mere substitute for it.

In particular, Fowler describes his use of an electronic concordance to conduct an intertextual analysis of the ending of book 10 of Silius' *Punica* (1997, pp. 20-24). He recounts how, while listening to a paper being given by a colleague on some aspect of Silius' poetry, a specific word triggers in him a Vergilian reminiscence. The word at issue is "maneres" and occurs in a passage with which Fowler was "excessively familiar

²⁶ On the nonlinearity of printed books see *infra* p. 54.

from teaching". After searching for the occurrences of this word in the PHI, another Vergilian passage where this word occurs is brought to his attention. Although he had "(shamingly) *not* remembered" this second parallel passage, he found it even more persuasive than the previous one.

However, the serendipity of computer search, which is aptly illustrated by Fowler's account, needs to be combined with a sound method for interpreting the results of automatic search. In the specific case of research on intertextuality, such a method needs to inform decisions especially in those cases where an automatic search suggests several possible literary models for a given line of text.

An example of such a methodology, within which the use of computer search can be situated, is provided by Smolenaars (2001). Well aware of the challenges related to the use of automatic search for the detection of allusions, Smolenaars devised a protocol aiming to guide the scholar through the exploration of possible intertextual parallels. This protocol, which is made up of five steps, allows for narrowing down the scope of the texts to be searched and provides a theoretical framework for the interpretation of the results returned by automatic searching.

2.3 BEYOND ELECTRONIC CONCORDANCES

A considerable part of the research carried out over the last decade in the field of Digital Classics can be seen as aiming to overcome the limitations of electronic corpora and concordances and, at the same time, aiming to add new tools for the study of texts to the classicist's toolkit.

In this section I discuss such tools and developments as they constitute the context within which my system to extract canonical citations was developed. Particular attention is devoted to tools for the automatic detection of intertextual parallels. Indeed, the automatic indexing of canonical references, which allows for retrieving parallels already discussed within secondary literature, can be seen as performing a complementary function to the discovery of new possible intertextual parallels.

2.3.1 *The Critical Apparatus of Digital Editions*

A characteristic of the TLG that has been almost unanimously criticised by classicists is its “flattened out” representation of the text (Ruhleder, 1995, p. 47). This flattening out is due, first, to the lack of a critical apparatus and, second, to the fact that the text of each work contained in the corpus is based on a single critical edition following the so-called “best edition approach”.

The greatest obstacle to having digital editions with critical apparatus has been to date the cost of encoding manually the wealth of information that critical apparatuses contain. Despite the recent attempt by Boschetti (2007) to address the automatic extraction of information from critical apparatuses, much work remains to be done in this area.

The method developed by Boschetti consists of parsing variants and conjectures from an Optical Character Recognition (OCR) transcription of a critical apparatus and mapping them to the reference text by means of alignment algorithms. This task is further complicated by the OCR errors caused by the small font-size in which critical apparatuses are usually printed and the mixture of Greek and Latin scripts that characterises them.

2.3.2 *Digital Editions of Quoted Texts*

Another limitation of the TLG that recent studies have addressed is how quotations are represented within the corpus. The standard for text encoding available at the time when the TLG was created – i.e. Betacode – did not allow for marking explicitly quotations of other texts in order to distinguish them from the surrounding text.

Such an ability, however, plays a key role in the creation of digital editions of fragmentary and gnomological texts. Fragmentary texts are ancient works that got lost and are known to us only as quotations contained in other sources. Gnomologies – or wisdom literature – are collections of textual materials ranging from pithy sayings to short excerpts of philosophical texts.

Technologies such as the markup scheme defined by the Text Encoding Initiative (TEI), the Canonical Text Services (CTS) and formal ontologies have been combined together and applied to the encoding of quo-

tations in the context of fragmentary and gnomological texts.²⁷ Recent studies on digital editions of fragmentary texts have employed these technologies to address the problem of representing the multiplicity of interpretations that fragments entail. In fact, scholars often disagree as to where in the text the quoted fragment starts and where it ends (Romanello et al., 2009b; Trachsel, 2012; Almas and Berti, 2013; Berti, 2013).

The same set of technologies were used also in the recently completed Sharing Ancient Wisdoms (SAWS) project²⁸, which focussed on gnomological texts. A challenging aspect of working with these texts is how to represent within a digital edition the relations between the source text and the passages extracted from it as the transmission of these passages may have gone through several historical periods and languages. The SAWS ontology is particularly relevant in the context of this research as it defines a formal taxonomy of quotations and other relations that may be identified between text passages. Examples of the relations modelled by this ontology are *verbatim* (i.e. literal) citations, paraphrases or translations.²⁹

2.3.3 *Tools for the Study of Intertextuality*

As discussed above, electronic concordances are suitable to identify a limited subset of intertextual parallels, namely those consisting of an almost exact repetition of words. Moreover, they require the user to know in advance what to search for – that vague memory triggering the search as in the anecdote reported by Fowler that was discussed above.

Recent research on the digital study of intertextuality has attempted to address these two limitations. First, by developing methods and tools supporting the discovery of intertextual relations where the similarity is determined by elements other than lexical similarity (e.g. the use of the same metrical structure, a reference to a common theme or the use of words with the same meaning). Second, by exploring ways to elicit

²⁷ In particular, the use of CTS and formal ontologies to represent canonical references is discussed in section 3.5.

²⁸ Sharing Ancient Wisdoms, <http://www.ancientwisdoms.ac.uk/>.

²⁹ For a discussion of the technical challenges raised by gnomological texts that were tackled by SAWS see in particular Tupman et al. (2012); Jordanous et al. (2012a,o). Further publications on other aspects of SAWS are listed in the dedicated page of the project website, <http://www.ancientwisdoms.ac.uk/publications>.

automatically similarities from two or more texts in order to cover those cases when it is not possible – or desirable – to specify in advance what to search for.³⁰

Musisque Deoque

The first limitation is tackled by Musisque Deoque³¹, a searchable electronic corpus that was deliberately conceived to support the study of intertextuality (Manca et al., 2011). Musisque Deoque is a database of Latin texts characterised by a very wide breadth as it spans from the origins to the Italian Renaissance. Such a breadth reflects the main aim of this resource, which is to support the study of diachronic intertextuality and literary influences. The texts contained in Musisque Deoque intentionally lack a critical apparatus but, instead, are provided with a collation of significant variants attested within the manuscript tradition.³²

The search facility offered by Musisque Deoque enhances the functionalities of an electronic concordance in two respects. First, it allows the user to search for a given expression not only within the established text but also within the repertory of variants. This innovative feature is essential to study literary influence as variants that may not deserve to be included in a critical apparatus may nevertheless reveal some interesting aspect of the tradition of a text. Second, the user can decide to apply some metrical filters to their query when searching through the texts. These filters allow for filtering the results based on their metre (e.g. hexameter), which is particularly useful when working with poetic texts.³³

Automatic Detection of Text Reuse

The second limitation of electronic concordances in supporting the discovery of intertextual parallels – i.e. finding similarities between texts by means of non-targeted search – has been addressed by research car-

³⁰ This section has greatly benefited from the discussion that took place at the panel “Rethinking Text Reuse as Digital Classicists”, which was held at the Digital Humanities (DH) 2014 conference, see <http://dharchive.org/paper/DH2014/Panel-106.xml>.

³¹ Musisque Deoque, <http://www.mqdq.it/>.

³² For the definition of *significant variant* used in the context of Musisque Deoque see Manca et al. (2011, p. 131).

³³ The collection of essays edited by Mastandrea and Spinazzè (2011) contains several contributions on the study of intertextuality that have benefitted from the use of Musisque Deoque.

ried out only over the last few years. This research has built upon and extended the method developed in the field of Computational Linguistics for the automatic detection of *text reuse*. The term text reuse refers to the meaningful reiteration of text – usually beyond the mere repetition of common language – and encompasses a number of different relations between texts such as translation, quotation, allusion and plagiarism.

Research on text reuse dealt with modern texts and focussed especially on lexical similarity and position within the document as features indicating the presence of text reuse. On the contrary, recent research on the discovery of specific types of text reuse such as intertextual parallels and imitative allusions within ancient texts – sometimes referred to as *historical text reuse* – has been exploring the use of other features, in addition to lexical similarity, to capture their presence.³⁴

Research in the area of *historical text reuse* began with the work by Lee (2007) on the detection of textual similarities between the three synoptic Gospels. He developed a model relying on lexical similarity and position within the document that can be trained to capture text reuse within these texts based on/according to the hypothesis of (nine) different scholars.

Bamman and Crane (2008) have focussed on the discovery of what they called *imitative textual allusions* in a corpus of Latin texts. Their method is innovative insofar as it performs the detection based on features such as word order and syntax in addition to lexical similarity.

Furthermore, Bamman and Crane (2009) have tackled the specific issue of detecting two different kinds of text reuse, namely allusions and translations, in texts written in multiple languages. They evaluated their method against the task of finding the allusions to the *Aeneid* contained in John Milton's *Paradise Lost*. Their evaluation shows that this task is particularly challenging because it entails the alignment of texts that, being written in different languages, follow different word orders and offer no fixed boundaries for the detection (e.g. sentence boundaries).

Finally, Büchler et al. (2012) have investigated the specific challenges in terms of computation that are raised by detecting text reuse on a large scale. In their study they have used an “all-against-all” method to compare with each other all the texts contained in the TLG. Moreover, they have proposed two measures to characterise the work of an author,

³⁴ See Büchler (2013), and especially pp. 55–86, for a discussion of the different challenges raised by *historical text reuse* as opposed to *text reuse* on modern texts.

reuse temperature and reuse coverage. The former captures the density of text reuse within a given work or passage, while the latter characterises works that were re-used frequently across different chronological periods.

Tesserae

Among the recent developments in this area, *Tesserae*³⁵ deserves some particular consideration. In fact, it is to date the only freely available tool that spun off from the theoretical research discussed above.

Tesserae uses primarily lexical similarity to identify text passages that may contain an intertextual parallel (Coffee et al., 2013). What constitutes lexical similarity in this case is both the presence of identical words and the presence of words with identical lemma (i.e. dictionary headword). The accuracy of the tool was tested on the task of finding parallels between the first book of Lucan's *Bellum Civile* and the entire *Aeneid*. To perform the evaluation Coffee and collaborators created a benchmark set by collating the parallels identified by a number of commentators. The results of the evaluation showed that the lexical similarity that *Tesserae* captures is sufficient to detect approximately 70% of these parallels. The *Tesserae* team are currently developing and testing further methods to cover the remaining 30% of parallels that the tool is unable to detect. These methods include ways to capture metrical and phonetic similarity, the use of aligned vocabularies to capture semantic similarity between Greek and Latin texts and the use of word clustering techniques such as Topic Modelling to capture thematic similarity.³⁶

An aspect of *Tesserae* that makes it especially relevant in the context of this research is the possibility of relating the parallels that the tool identifies automatically to the secondary literature citing the same set of parallels. This kind of information could be gathered by mining canonical references from an archive of articles such as JSTOR and presented to the user in order to contextualise the search results. Alternatively, the existence of articles that cite a given set of parallels could be taken into account by the tool while ranking the candidates that may constitute intertextual parallels.

³⁵ *Tesserae*, <http://tesserae.caset.buffalo.edu/>.

³⁶ Some of these functionalities are available for testing on the *Tesserae* website at <http://tesserae.caset.buffalo.edu/experimental>.

2.4 SUMMARY

The analysis of references to parallel passages as found in a diachronic sample of commentaries shows that their appearance and structure stabilise by the end of the XIXth century. In contrast, their rhetorical function evolved substantially over the last two centuries. It is possible to observe, in fact, an increasingly common tendency to provide lists of parallel passages, often introduced by the formula “cf e.g.”, without further clarifications of how they contribute to or advance the interpretation of the commented text to which they are juxtaposed. Although the relation between this tendency and the digital tools – first and foremost electronic corpora and concordances – remains to be proven, there is little doubt that such tools have changed the *modus operandi* of classicists, especially with regards to the retrieval and discovery of parallel passages. Once the automatic indexing of canonical references is added to the classicist’s toolkit, scholars will be able to search for literature discussing specific sets of parallels in addition to searching for new possible intertextual parallels, a function that is provided by recently developed tools.

DISENTANGLING CANONICAL CITATIONS: FROM HUMAN-READABLE NOTATION TO FORMAL MODEL

Overview

In this chapter I describe the process of transforming canonical references from a human-readable *notation* into a machine-understandable *model*. In section 3.1 I justify the need for this model in the context of the automatic extraction of such references. In section 3.2 I present an analysis to identify the main characteristics of canonical citations that need to be modelled. I then introduce in section 3.2 the key concepts related to the creation and publication of ontologies – a particular kind of formal model. In section 3.5 I discuss the creation of an ontology of canonical references called HuCit. Finally, in section 3.6 I consider how a database underpinned by HuCit is used to support the automatic extraction of canonical references.

3.1 TRANSFORMING A HUMAN-READABLE NOTATION INTO A FORMAL MODEL

In this section I justify the need for a formal model of canonical citations with regards to the automatic extraction of such citations from text.

Canonical citations are artefacts that are meant to be understood and consumed by humans, not machines. As such, they rely heavily on contextual information as well as on the capabilities of the trained reader to disambiguate or to fill in information that was left intentionally implicit. On the contrary, making citations computationally tractable means having to make explicit information that, when writing, can easily be left implicit or unspecified without compromising the meaning or intelligibility of the reference.

In other words, in order to transform citations from a human understandable notation into a computable set of relationships between texts (or portions of them) it is necessary to create a model of what is being cited. Such a model is characterised by a “demand for computational tractability, i.e., for complete explicitness and absolute consistency” and by the “manipulability of a computational representation” (McCarty, 2004).

The Humanities Citation Ontology (HuCit) described in section 3.5 is the formal model of citations I devised to support the automatic extraction of citations. This model is a formalisation of the underlying model that the practice of citing texts already implies and reflects. Indeed, as Smith (2009) aptly put it, “citation is already a form of ontology”, where he uses the term *ontology* in the philosophical sense to mean a theory concerning the existence of things. A text passage must exist in order to be cited and this also holds true for other citable objects such as coins, inscriptions, papyri, etc. How we refer to a text tells us something about how we perceive its structure and properties.

While models are necessary for computation, any model is inevitably going to be an approximation of reality. Arguably, even the practice of canonical citations itself is based on an idealised view of texts. Greek and Latin works are treated as a frozen and thus stable corpus of texts for the practical purpose of making them citable. Aspects that tend to get simplified and therefore hidden in a citation system are those that are most disputed by scholars as the following examples illustrate. Scholars, for instance, may cite the *Heroides* 15 as Ov. *Herod.* 15 for pure convenience, while the question of whether the authorship of this poem should be attributed to Ovid remains debatable. Similarly, some scholars may cite Propertius 2.22, while others, who do not believe in the unity of this elegy, may refer to it as 2.22a and 2.22b.

In the context of the automatic extraction of canonical citations the use of a formal model responds to two main needs, i.e. the need for explicitness and the need for domain knowledge.

The Need for Explicitness

An example of the need for a formal, explicit model is offered by citations that span several sections of a text (e.g. Plato *Rep.* 595a–596a). A human reader does not need to know which sections are comprised within this range in order to locate the cited passage. However, when

creating an electronic index this information has to be made explicit in order to become searchable. A query for occurrences of Plato *Rep.* 595b and 595c will also return the document citing *Rep.* 595a–596a but only if the implicit reference is expressed in the underlying model (figure 3.1).

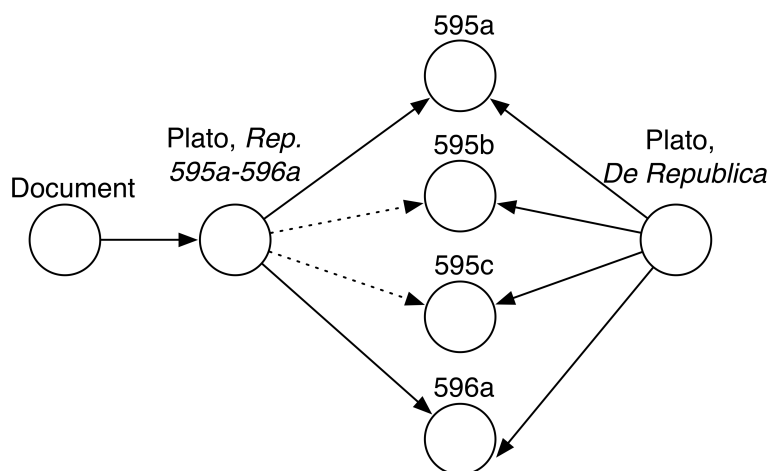


Figure 3.1: This graph expresses a reference pointing to a specific section of Plato's *De Republica* – Plato *Rep.* 595a–596a. The references to sections 595b and 595c, which are implicitly contained in the range 595a–596a, need to be made explicit when modelling this canonical reference and thus are indicated by dotted lines.

There are, however, citations that cannot be made computationally tractable without removing the vagueness that natural language allows for. The citation “Hom. *Il.* I 5 ss.”, for example, is read as pointing to book 1, line 5 and the following lines of Homer's *Iliad* (represented by “ss.”). The author indicates where the citation starts – at line 5 – but leaves intentionally unspecified where the citation ends. It is up to the reader to decide the precise meaning of this reference. When a formal model of this citation is created it is necessary to specify where the citation starts and where it ends. Therefore it becomes problematic to retain the vague reference to the following lines. In a formal model citations have to be either punctual – that is, referring to one single section of a text, whatever this may be – or range-like, i.e. having a start and an end. My approach to this problem has been to express such open-ended ranges by means of a search parameter that the user can set. For example, setting this parameter to 10 causes the citation “Hom. *Il.* I 5 ss.” to be treated as if it were “Hom. *Il.* I 5–15”. This solution makes a vague citation computationally tractable while maintaining its original vagueness in the form of a parameter that can take different values.

The Need for Domain Knowledge

Another important function of such a model is to provide the citation extraction system with a surrogate of domain knowledge that is necessary to correctly interpret canonical citations. Such knowledge cannot be deduced or inferred from the surrounding context of a citation and is acquired by human readers by means of training.

The citation “Thuc. 1.22.4” is a fitting example of the domain knowledge required to understand fully the meaning of this citation. It refers to the famous passage of the *Histories* where Thucydides defines his work as a “possession for all time” (κατὰ πάντα τοῖς αἰεὶ). The abbreviation of the work’s title can be omitted as the *Histories* is considered Thucydides’ *opus maximum* – i.e. the only work produced by a given author or, in the case of multiple works, the most known – and it is common practice in such cases to provide only the abbreviation of the author’s name, followed by the indication of the passage being cited.

This sort of implicit information or domain knowledge, however, is not accessible to a computer programme unless it is explicitly encoded in a model and store in a *knowledge base* that can be used as a basis for reasoning and inferencing. A knowledge base built upon HuCit is described in section 3.6 and its use to disambiguate automatically extracted citations is discussed in section 4.4.4.

I turn now to analyse in more detail the characteristics of how classicists refer to ancient texts. Particular attention will be paid to how such canonical citation practices emerged and evolved over time.

3.2 UNDERSTANDING CANONICAL TEXTS

As argued in the previous section, citation practices reflect a view of the specific domain to which the cited texts belong. The construction of a formal model of such canonical citations needs to start from the analysis of the domain of interest, which is the object of this section.

I first start by explaining the importance of an agreed upon way of citing texts as a requirement for building any scholarly discourse about those texts (section 3.2.1). Then I consider some selected examples of canonical citation schemes of classical works with the goal of reconstructing and tracing their origins (section 3.2.2). I conclude by sum-

ming up the main characteristics of canonical citations that I had to deal with when building HuCit (section 3.2.3).

3.2.1 *Canonical Citations and their Importance for a Scholarly Discourse About Texts*

The system of canonical citations is a remarkable example of interoperability of text references across virtually all editions and translations of the same work. Its importance lies in the fact that it is mandatory when building a scholarly discourse about texts. In some areas of Classics, such as classical philology, the ancient texts are not only sources of information but, in fact, the very object of research and as such it is essential that they are referenced in a common, precise and stable way.

From a pragmatic point of view, there are two conditions that need to be satisfied for the system of canonical citations to work. First, they need to refer to a canonical division of a text, the *canonical citation (or numbering) scheme*. Second, references to such a citation scheme also need to be provided within editions and translations of a text to allow the reader to look up a given citation.

In order to guarantee that these two conditions are satisfied, classicists are trained at a very early stage of their career to cite texts appropriately. This is reflected by the numerous resources available that contain guidelines for citing ancient texts. In addition, the care devoted to how texts are cited is an element that is often taken into account in book reviews, for example, when assessing the overall quality of a publication.

A quick search through the electronic archives of the Bryn Mawr Classical Review (BMCR) returns several examples of both positive and negative remarks about the reviewed publication that specifically address the behaviour of the authors or editors towards citing primary sources.

The use of Bekker numbers¹ in the margins of a translation of Aristotle's *Nicomachean Ethics* is praised as a "mechanical benefit" of the edition reviewed in BMCR 2001.09.24² that makes it "much easier to cite and more useful for those who wish to use it with the Greek original close to hand". On the contrary, the absence of such references from a translation of Aristotle's *Politics* reviewed in BMCR 98.1.21³ is lamented

¹ On the use of Bekker number to cite Aristotle's works see *infra* p. 58.

² BMCR 2009.07.24, <http://bmcr.brynmawr.edu/2009/2009-07-24>.

³ BMCR 98.1.21, <http://bmcr.brynmawr.edu/1998/98.1.21>.

by the reviewer as the cause of the inability of students “to cite passages from it with precision or compare this translation with others”.

Moreover, the reviewer of an edited volume on Socratic studies in BMCR 1998.10.11⁴ points out in footnotes 6, 11 and 12 that it would have been helpful if the authors had followed a more precise method to cite Plato, that is with the indication of line numbers in addition to Stephanus page and letter – i.e. the canonical way of referring to Plato’s works. A common trait of all these reviews is that they stress the importance of how texts are cited as an essential aspect of a rigorous philological method.

3.2.2 *Canonical Citation Schemes as Historical Objects*

We take the structure and divisions in the classical texts we encounter today for granted. However, the creation of the HuCit ontology of citations required a closer inspection and definition of the nature of these text divisions in order to model and understand the historical process underlying them. This analysis is presented and discussed in the current section.

A Brief History of Canonical Citation Schemes

The division of a text into smaller chunks, for easier reading and citing, is the result of a historical process which, in turn, is closely connected with the history of the transmission of ancient texts. This process responds to specific needs – such as the need of a scholarly community for precise methods to refer to its primary sources – and, at the same time, is deeply affected by technological innovation in writing and reading support (Kalvesmaki, 2014).

In antiquity, the division into books as a way of organising and editing texts dates back to the Hellenistic period, and specifically to the work of the scholar-librarians of Alexandria (Higbie, 2010). There are earlier attestations of the use of book divisions, but solely as an organising principle in the composition of new texts. By the beginning of the 3rd century CE the practice of citing sources by book number had replaced, albeit not completely superseded, the old practice of referring to specific scenes (e.g. “the struggle of Achilles”, “the death of Hector”, etc.).

⁴ BMCR 1998.10.11, <http://bmcr.brynmawr.edu/1998/1998-10-11>.

However, it is only with the transition from the volumen to the codex that more granular ways of dividing and referring to texts start to emerge. The revolutionary element introduced by the codex is the page, which has the effect of breaking the linearity of the text into units of smaller size.⁵ In the 4th century CE this nonlinearity together with the need of clergy for a more convenient way of finding passages from the Scriptures were the conditions for the invention of one of the first canonical numbering schemes, i.e. the numbered division of the New Testament gospels into *titloi* and *kephalaia* devised by Eusebius of Caesarea (Kalvesmaki, 2014, para. 8).

The division of the text into even more granular units such as subsections that make reading more pleasant and citing easier coincides with another technological innovation, the transition from the manuscript to the printed book during the Renaissance (Febvre and Martin, 1958, p. 128). Examples of text divisions from this period that later became canonical citation schemes for specific works are the Stephanus pagination for citations of Plato and the one by Casaubon for citations of Athenaeus, both discussed later in this section. Also, it is in this period that the use of page numbers becomes common practice among printers, thus enabling more precise references to specific text sections.

Moving forward to the present time, the transition to yet another medium – the electronic text – urges us to re-conceptualise canonical citation schemes. Digital technologies enable new and more sophisticated ways of referring to specific portions of the text. At the same time, they allow us to align different citation schemes one to another, which is of essential importance especially for those texts for which multiple competing citation schemes exist.⁶ This is the case for example with Aristotle's works that can be cited by Bekker numbers as well as by book, chapter and sentence.⁷

The process of encoding a text to produce a digital edition forces us to reflect on such canonical citation schemes and to encode them so as to

⁵ The transition from the volumen to the codex and its consequences are often discussed in the recent literature about the hypertext as an early example of the nonlinearity that characterises the hypertext as medium, see e.g. Vandendorpe (2009, pp. 28–39) and O'Donnell (1998, pp. 50–63).

⁶ See Kalvesmaki (2014) for a detailed discussion of the challenges posed by canonical citation schemes to the creators of digital editions. In particular, Kalvesmaki (2014, para. 45–54) contains some useful recommendations on this specific topic that are targeted to managers of digital edition projects.

⁷ See *infra* p. 58.

allow the users to keep citing texts as they already do without having to adapt their practice to the new medium. For example, book publishers are facing the issue of allowing readers of the electronic version of a book to find out the corresponding page number of the printed edition for a given passage. The page number – as opposed to other ways of locating text passages that ebook readers have developed – remains the system of choice to cite texts in academic publications.

I turn now to examine some selected examples that show how the canonical citation schemes that are currently employed to cite classical texts can be better understood if considered from a historical perspective.

Citing the Homeric Poems

Homeric poems are commonly cited by book number and by line (e.g. Hom. *Il.* 1.1). The division of the text into lines corresponds to its metrical structure (i.e. the hexameter), while the division into books is currently attributed to the Alexandrian grammarian Zenodotus – although the ancient sources attribute it to Aristarchus of Samothrace. Zenodotus may be responsible for dividing the text by copying it onto twenty-four papyrus scrolls numbered, as was common practice in antiquity, using the letters of the Greek alphabet.

The physical supports and media through which the poems were transmitted to us changed over time – from scroll to codex to printed book and, now, to the digital edition – but that division, which originally came from the actual distribution of text over several papyrus scrolls, became and remains canonical.

Traces of the Alexandrian division of the Homeric poems into books, and the resulting numbering, are still visible in one specific style of referring to those poems. This style uses the uppercase letters of the Greek alphabet to identify the books of the *Iliad* and the lowercase ones for the *Odyssey* (e.g. α 1 stands for Hom. *Od.* 1). Because of its conciseness this style is primarily used in publications focussing on the study of epics, which often contains hundreds of references to the homeric poems.

The division of text into lines is common for works of poetry as it is based on the prosody and constitutes a fine-grained structure of the text that in most cases becomes also canonical. Works of prose, on the contrary, tend to be longer and lack this division into smaller chunks.

For this reason editors and printers had to find a solution to make them easier to read and cite, as the following examples illustrate.

Citing Plato

The works by Plato are cited according to a citation scheme that derives from the edition of the text by Henri II Estienne – also known as Stephanus (e.g. Plato, *Nomoi* 835c–842a).

Stephanus, who was member of a dynasty of humanist scholar-printers active in Paris and Geneva between the 16th and 17th centuries, was one of the most important humanists and philologists of the French Renaissance and printed hundreds of editions of Greek and Latin authors (Jehasse, 1976; Kecskeméti et al., 2003). He is an especially interesting figure as he combined – as many scholar-printers of that period did – a philological attention to establishing a text that was as close to the original as possible with the attention to the typographical details of the printed edition.

The current canonical way of citing Plato's text is based on the pagination of Stephanus' edition of the works of Plato that was published in three volumes in 1578. A typical citation contains the work title, often abbreviated, followed by the page number of the Stephanus edition, followed by a Latin letter from "a" to "e" that refers to the division of each page into sections of roughly the same length. However, a more precise method to refer to Plato's text consists of adding to the citation the indication of the line number, taking as reference the line numbering of a more recent critical edition – i.e. the edition published by Burnet between 1900 and 1907 in the Oxford Classical Texts series.

Stephanus' edition looks modern and not very different from those that are printed nowadays, but it was novelty in its time. His edition of Plato was the first to show Greek text and Latin translation side-by-side, surrounded by an outer apparatus of notes.⁸ Moreover, what feels revolutionary about his edition of Plato is the way in which the space of the printed page is organised. The clarity in the distribution of text, together with reading aids such as the section letters printed in the margin, make for a more easily readable and citable text.⁹ Even the use

⁸ A previous edition by Valder (1534) only had the Greek text surrounded by Proclus' commentaries, and the commentaries were completely removed from an edition published in Basel in 1556 in order to reduce the costs of printing (Sellars, 2013).

⁹ Concerning the editorial task of text division, Stephanus probably learned the lesson of his ancestors. The modern versification of the New Testament is due to his father,

of pagination was, at the time of Stephanus, a relatively recent introduction. Its first appearance is most probably in a book published by Aldus Manutius in 1499, but its use became systematic only in the second quarter of the 16th century thanks to the humanist printers (Febvre and Martin, 1958, p. 30).

Citing Athenaeus and Aristotle

Citation systems similar to the one that was just described – i.e. based on the page number and section letter of a certain edition – are used to refer to the works of Athenaeus and Aristotle.

The edition of Athenaeus' *Deipnosophistae* published by Isaac Casaubon in 1598 presents an organisation of the printed page similar to Henri Estienne's edition of Plato.¹⁰ This led to an analogous way of citing the text consisting of a mixture of arabic numbers indicating the pages and letters for the page sections (e.g. Athen. *Deipn.* 556f).

The current way of citing Aristotle's works consists of using references to page, column and line numbers of the 1831 edition by the German philologist Immanuel Bekker (e.g. Arist. *Pol.* 1304a 17–24).¹¹ The importance of this edition is due to the reorganization of the entire *corpus aristotelicum* that Bekker carried out.

What is particularly interesting about citations of Aristotle is the co-existence in current practice of two different citation schemes, i.e. Bekker pages on the one hand and the medieval division into books, chapters and sentences on the other. Their co-existence is due to the fact that each of these schemes serves a different purpose. The former is better suited to refer precisely to a specific line – or even word – of the text, whereas the latter favours a division of the narrative into logical segments such as chapters and sentences (Kalvesmaki, 2014, para. 24).

Robert Estienne, while his grandfather Henri I Estienne was responsible for a verse numbering of the Psalms that did not catch on. For a discussion of their contribution to the modern division of the Bible see Zola (2012), especially pp. 244–246.

¹⁰ The date of publication of Casaubon's edition as well as the fact that he was close to Estienne, having married Florence the daughter of Henri Estienne, suggests he might have been inspired by Henri Estienne's lesson with respect to typography and layout.

¹¹ Barnes (1995, p. xxi) aptly explains Bekker's numbers as follows: "The reference, 'HA A 6, 491a9–14', first gives the title of the work in question ('HA' abbreviates 'History of Animals'); then the book number (the 'A' here is a Greek alpha and refers to the first book of the *History*); then the chapter number (the Arabic numeral '6'); and finally the Bekker code: page number, column number, line numbers (here, lines 9 to 14 of the left-hand column on the four hundred and ninety-first page)".

3.2.3 *Main Characteristics of Canonical Citation Schemes*

The examples discussed in the previous section give an idea of the variety of ways in which citation schemes came into use. If a classification were to be made, canonical divisions could be grouped into the following categories:

1. Canonical schemes that derive from features of the content (e.g. prosodic structure);
2. Canonical schemes that derive directly from the characteristics of one of the physical supports on which the text was transmitted to us;
3. Canonical schemes that correspond to the original logical structure of the text, given to it by its author or by the authority of tradition;
4. Canonical schemes that are inherited from the physical structure of the text as it appears in one specific edition, chosen for instance because of its importance or novelty.

These categories are not mutually exclusive as many of the examples discussed in the previous section belong to more than one category.¹² The citations of homeric poems fall into categories 1 and 3: in fact, the division into lines derives from the meter in which the poems were composed, whereas their division into books was a deliberate decision of the Alexandrian scholars. Moreover, citations of Aristotle that provide the book number in addition to the Bekker page follow a citation scheme belonging to both 3 and 4.

The main properties of canonical citation schemes can be summarised as follows:

- they define flat or hierarchical ordered structures of text elements that can be cited;

¹² I propose such a categorisation as an alternative to the clear-cut distinction of citation schemes into the opposing categories of *logical* and *physical* (Smith, 2009) or *semantic* and *visual* (Kalvesmaki, 2014). According to this distinction, the division into books/chapters/sections belongs to the former, while the page numbering belongs to the latter.

- they emerge as the result of the activity of editors and publishers, and their success – i.e. the fact of becoming canonical – depends on the acceptance and use by the rest of the community;
- they evolve over time and are sensitive to the changes in the technology of reading and writing supports;
- they are not unique, as several citation schemes to refer to the same text may co-exist, each reflecting a different interpretation of that text and performing a specific function.

3.3 KEY CONCEPTS OF ONTOLOGIES AND SEMANTIC WEB

3.3.1 *What is an Ontology?*

Definition

In philosophy the term ontology is used to refer to “that branch of metaphysics concerned with the nature or essence of being or existence” (*Oxford English Dictionary (OED)*, s.v. ‘ontology’). However, this term is used in this dissertation with the meaning it acquired in Computer Science. In this sense, an ontology is a computational artefact which provides a formal model of a domain of interest consisting of classes of objects, their attributes and the relations between them.¹³

Another widely accepted definition of ontology is the one by Gruber, who defines an ontology as an “explicit specification of a conceptualization” (Gruber, 1995). Gruber builds upon the notion of conceptualisation as “the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them” (Genesereth and Nilsson, 1987).

A refinement of Gruber’s definition was proposed by Giarretta and Guarino (1995) who define an ontology as an explicit formalisation of a *shared* conceptualisation (emphasis my own). Their definition emphasises how the benefits of creating ontologies that are not based on shared conceptualisations are limited as they are not reusable or interoperable.

An ontology contains classes that describe concepts in the domain of interest. Classes are related to each other by means of taxonomical

¹³ For a rigorous discussion of the various meanings of the term ontology see Giarretta and Guarino (1995); Smith (2003); Guarino et al. (2009).

relations, i.e. subclass and superclass relations. In addition to such taxonomical relations classes define properties that are common to all instances of a given class. These properties can take as their value an instance of another class.

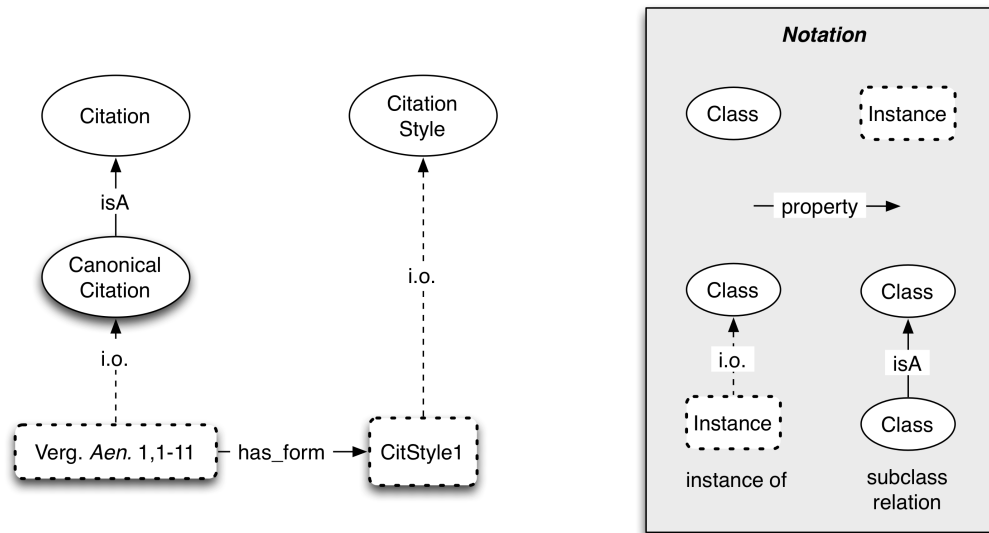


Figure 3.2: The symbols used in this chapter to illustrate the structure of the ontology. The symbols and their meaning are shown on the right, while an example of their usage is provided on the left.

Consider the example shown in figure 3.2, which introduces the visual notation employed in the rest of this chapter when describing classes and properties of an ontology. The class *CanonicalCitation* is a subclass of *Citation* as it describes a concept that is more specific than its superclass.¹⁴ Moreover, two instances are defined: “*Verg. Aen. 1,1-11*” is an instance of the class *CanonicalCitation* and “*CitStyle1*” is an instance of *CitationStyle*. The fact that the canonical citation follows a specific citation style is expressed using the *has_form* property defined by the class *CanonicalCitation*.

Types of Ontologies

Ontologies can be classified based on various criteria, such as their degree of formalisation (light-weight and heavy-weight ontologies) or their degree of generality and reusability outside their target domain, in which case they can be divided into the following groups¹⁵ (here in inverse order of generality):

¹⁴ A fixed space font is used throughout this chapter whenever ontology classes and relations are referred to.

¹⁵ See Staab and Studer (2009, p. xii).

- *foundational ontologies* (also called top-level or upper-level ontologies) define classes and relations that are general and domain-independent (e.g. events, space, time, etc.);
- *domain ontologies* are designed to model phenomena that are relevant for a specific domain, thus reflecting the conceptualisations and viewpoint of users in that specific domain;
- *task ontologies* define classes and relations that describe a specific task or activity (e.g. text editing or annotation);
- *application ontologies* specialise domain and task ontologies to meet the purposes of a specific application.

HuCit can be considered a domain ontology as it models the domain of canonical texts and citations from the perspective of classicists. Moreover, the ontology extends other domain ontologies such as the CIDOC Conceptual Reference Model (CIDOC CRM) and FRBR Object-Oriented (FRBR_{OO}), which are both reviewed in section 3.4.1. CIDOC CRM, in particular, is an interesting case as it falls into the categories of domain *and* foundational ontology. In fact, it describes information in the domain of cultural heritage while defining a core set of very general categories – i.e. space-time information, events, material and immaterial things – that other domain ontologies can build upon.

Ontology and Knowledge Base

A distinction often made in this chapter is the one between ontology and knowledge base. The difference is that the former contains the definitions of classes and properties that form an ontology, while the latter contains a set of instances of the classes defined in the underlying ontology.

Applying this distinction to the example introduced in the previous section, the classes *Citation*, *CanonicalCitation* and *CitationStyle* are part of the ontology, whereas all instances of these classes belong to the knowledge base. More precisely, this distinction corresponds to the distinction drawn in Descriptive Logics between the *terminological component* (TBox) and the *assertion component* (ABox), which represent respectively the intensional and extensional aspect of an ontology.

A knowledge base is one of the main components of an *expert system*, a system which imitates an expert in a given field by applying a

form of knowledge-based reasoning to solve a specific set of problems (Feigenbaum and Klah, 2003). The knowledge base stores the body of structured information used by the reasoning or inference engine, the other main component of an expert system.

The goal of HuCit and the knowledge base built on it is to support the expert system for the extraction of canonical citations described in chapter 4. Indeed, the facts and assertions that are stored in the knowledge base provide a surrogate of domain knowledge which is needed, for example, to disambiguate correctly the citations once extracted.¹⁶

3.3.2 *Methods for Building Ontologies*

Ontology engineering and ontology learning are the two main methodological approaches to constructing ontologies. Although they differ substantially from each other, they are not mutually exclusive and they often appear together at different stages of the formalisation of ontologies.

Ontology Engineering

This approach consists of the manual construction of an ontology. It is typically carried out by an ontology or knowledge engineer on the basis of information about the target domain. This information is gathered by means of interviews with domain experts in a so-called *domain analysis*. Efforts have been made over the years to define clear scientific principles to be applied to this task such as the work by Mizoguchi (2004), the OntoClean methodology (Guarino and Welty, 2009) or the attempt to define a set of Ontology Design Patterns (Presutti and Gangemi, 2008).¹⁷

Ontology Learning

In contrast this approach aims at the automatic or semi-automatic construction and population of ontologies and predominantly relies on machine learning and Natural Language Processing (NLP) techniques to achieve this goal. In fact the term *ontology learning* stands for a wider range of subtasks, often referred to as the *ontology learning layer cake*,

¹⁶ Some practical examples of the reasoning that such a knowledge base allows for are described in section 3.6.

¹⁷ For an overview see Sure et al. (2009).

namely the extraction of terms, synonyms, concepts, taxonomic relations, non-taxonomic relations and axioms (Buitelaar and Magnini, 2005).

Ontology learning is an area of research that has only been explored in the last 15 years and has received a substantial boost in publications recently.¹⁸ The main advantage of this approach over ontology engineering is that it can drastically reduce the time required to build ontologies, a real bottleneck in such endeavours. This approach applies techniques that are mostly drawn from fields of research where the accuracy of the applied algorithms can be measured against precise metrics. As a result, it inherits from them the notion of measurable quality of output. The evaluation of existing techniques is proving to be, however, one of the big challenges in the area of ontology learning due to the difficulty of assessing the quality of knowledge representations (Cimiano, 2006).

The main assumption underlying this approach is that knowledge regarding a given domain is already verbally expressed in texts in a way that is suitable to be captured automatically – to some extent and with some degree of accuracy – by means of NLP techniques. This approach essentially requires the texts used for the ontology learning process to reflect a structured, explicit, logical representation of the domain. This may or may not always be the case, as practitioners may think and thus write, about their own domain without having in mind a clearly structured ontological view of it or leaving some information implicit. The main limitation of this approach lies in the fact that the overall quality of the extracted ontology depends on the accuracy of the tools and the techniques that are used as well as on the make-up of the texts upon which it is performed.

3.3.3 *Publishing Ontologies and Semantic Data on the Web*

The Semantic Web Vision

The Semantic Web was envisaged as “the evolution of a Web that consisted largely of documents for humans to read to one that included data and information for computers to manipulate” (Berners-Lee et al., 2001; Shadbolt et al., 2006). Unlike the Web of Documents, in the Web of

¹⁸ For a thorough review of the state-of-the-art of ontology learning technologies see (Wong et al., 2012), which renders as outdated the one contained in Cimiano et al. (2006).

Data information is expressed and published in such a way as to enable machines to manipulate it and to reason upon it.

Ontologies are a fundamental asset for the realisation of this vision as they define the terms that can be used to describe the underlying meaning of data in a machine actionable way. Moreover, equivalencies established between terms from different ontologies that describe the same concepts are essential in order to achieve interoperability between diverse sets of data published on the Web.

Semantic Web Standards

Over the last decade the World Wide Web Consortium (W3C) has been developing a family of standards and technologies to realise at a technical level the Semantic Web vision. At the core of these standards is the Resource Description Framework (RDF), a graph-based data model which uses subject-predicate-object statements called *triples* to describe semantic data. The Web Ontology Language (OWL) is a family of languages to describe and publish formal ontologies based on the RDF data model. Finally, SPARQL (SPARQL Protocol And RDF Query Language) provides a language for querying semantic data comparable to what Structured Query Language (SQL) does for querying relational databases.

Let us consider now in more detail the basic principles of RDF, given that this is the data model I used to specify the HuCit ontology as well as the knowledge base. As was already mentioned, RDF allows us to formulate statements about Web resources in the form of subject-predicate-object expressions called triples. Uniform Resource Identifiers (URIs) are used to identify the subject, predicate or object of an RDF triple. Figure 3.3 shows a graph representing three RDF triples drawn from the HuCit ontology. Nodes in the graph represent Web resources and are connected by edges representing the predicates of RDF statements. Since edges represent predicates their direction matters; they go *from* the subject of a statement to its object.

The triples in figure 3.3 can be read as:

- the resource identified by the URI <http://purl.org/net/hucit#Citation> has type `Class`, defined by the resource <http://www.w3.org/2002/07/owl#Class>. The property type is defined by <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>.

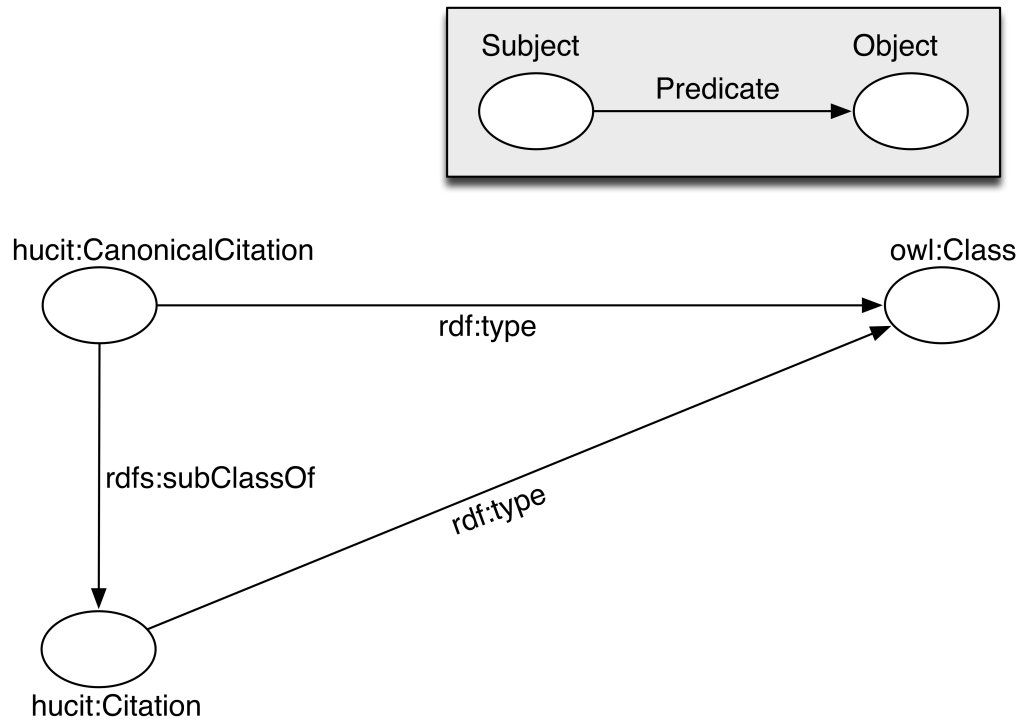


Figure 3.3: Graph representation of an RDF triple.

- the resource identified by the URI [http://purl.org/net/hucit# CanonicalCitation](http://purl.org/net/hucit#CanonicalCitation) also has type Class;
- the class CanonicalCitation is a subclass of Citation. The property subClassOf is defined by the URI <http://www.w3.org/2000/01/rdf-schema#subClassOf>.

Since RDF is a data model – not a data format – it can be expressed (i.e. serialised) using various data formats. The triples of figure 3.3, for example, can be expressed using an Extensible Markup Language (XML) syntax shown in figure 3.4 or a text-based syntax called Terse RDF Triple Language (Turtle) shown in figure 3.5.

Linked Open Data

Linked Open Data (LOD) is a set of best practices for publishing data on the Web in such a way as to make it possible to fully realise the potential of the Semantic Web (Heath and Bizer, 2011). The most common application scenario for the Semantic Web is one where software agents automatically aggregate information from several sources and perform reasoning and inferencing upon it. Unfortunately, as RDF *per se* is not sufficient to enable this, the LOD defines some standardised ways of es-

```

<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:hucit="http://purl.org/net/hucit#"
  ↪ xmlns:owl="http://www.w3.org/2002/07/owl#"
  ↪ xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  ↪ xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdf:Description rdf:about="http://purl.org/net/hucit#Citation">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description
    ↪ rdf:about="http://purl.org/net/hucit#CanonicalCitation">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description
    ↪ rdf:about="http://purl.org/net/hucit#CanonicalCitation">
    <rdfs:subClassOf
      ↪ rdf:resource="http://purl.org/net/hucit#Citation"/>
  </rdf:Description>
</rdf:RDF>

```

Figure 3.4: An example of RDF triples expressed using the XML syntax.

```

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix hucit: <http://purl.org/net/hucit#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

hucit:Citation
  a owl:Class .

hucit:CanonicalCitation
  a owl:Class ;
  rdfs:subClassOf hucit:Citation .

```

Figure 3.5: Example of RDF triples expressed using the Turtle syntax.

tablishing links between data and between the terms that describe this data. Such links make it possible to combine several RDF graphs into a connected global graph, the LOD cloud (figure 3.6).

The LOD principles are:

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When looking up a URI, provide useful information, using the standards (RDF, SPARQL Protocol And RDF Query Language (SPARQL)).
4. Include links to other URIs, so that they can discover more things.

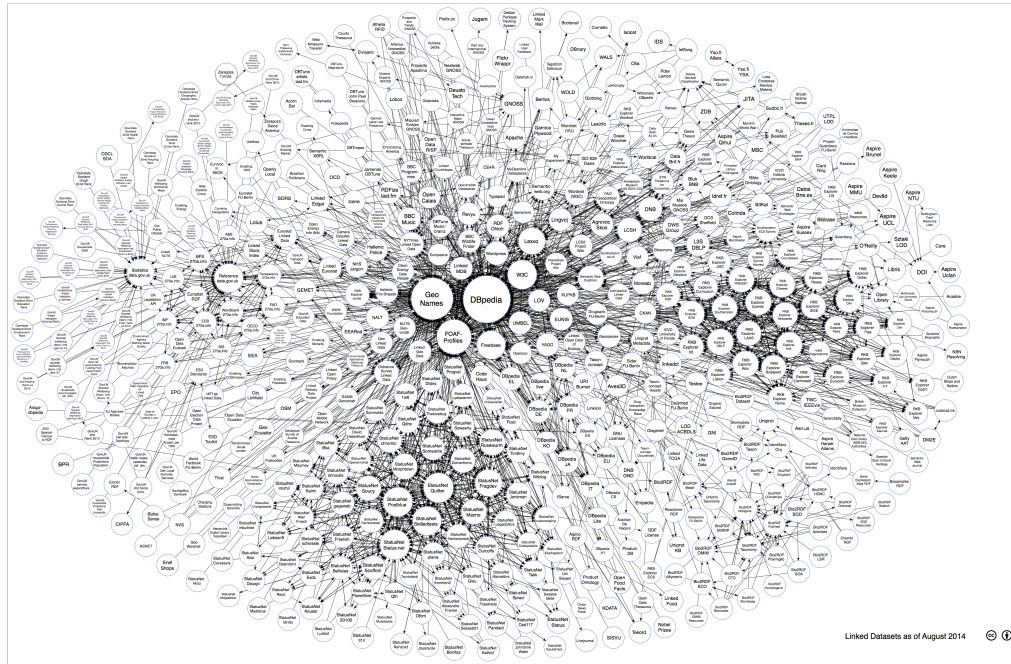


Figure 3.6: Linking Open Data cloud diagram 2014, by Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak. <http://lod-cloud.net/>

Three technologies provide the technical foundation of LOD: URIs, the HyperText Transfer Protocol (HTTP) and RDF. HTTP URIs are the names used on the Web to identify *things of interest* (or *resources*). It is worth emphasising that URIs identify abstract concepts rather than the documents that describe such concepts. The HTTP protocol allows clients to look up those names and to retrieve the corresponding RDF description – an operation called *dereferencing*.

Moreover, RDF links are essential in the LOD approach as they describe various types of relations existing between distributed resources as well as between vocabulary terms (i.e. ontology concepts). The property `sameAs` defined by OWL, for example, is used to assert an identity relationship between resources describing the same entity or concept.¹⁹

3.4 COMPUTATIONAL MODELS OF CITATIONS: A REVIEW

In this section I review the models of citations and bibliographic information that have informed the design of the HuCit ontology. I first discuss the Functional Requirements for Bibliographic Record (FRBR)

¹⁹ In section 3.4.3 I review ongoing efforts to advocate the adoption of LOD for publishing data about the ancient world.

model and its influence on the development of models of bibliographic information in Classics (section 3.4.1). I then examine how citations were modelled in the field of *semantic publishing* (section 3.4.2). I conclude by reviewing the underlying models of existing protocols to represent canonical citations as links (section 3.4.3).

3.4.1 *Conceptual Models of Bibliographic Information*

A citation expresses a relation between the citing and cited document. Creating an ontology of citations such as HuCit inevitably involves the formalisation of not only the nature of this relation but also the nature of the citing and cited documents. Existing models of bibliographic information that formalise the latter informed the design of HuCit and are discussed in this section.

FRBR

FRBR is a model of bibliographic information that has been widely accepted over the last decade in the area of Digital Classics, especially in relation to the creation of digital libraries of classical texts.

The creation of FRBR was one of the results of a process of formalising the conceptualisation that informs cataloguing rules. This process, promoted by the International Federation of Library Associations and Institutions (IFLA), started in the 90s and lasted a couple of decades (Willer and Dunsire, 2013). Such a process led to the definition of three models: FRBR, FRAD and FRSAD for bibliographic and authority data. These three models are the foundation of Resource Description and Access (RDA), a cataloguing standard for anglophone libraries. Of these three models the first one – FRBR – is especially relevant in the context of HuCit.

The importance of FRBR lies in its ability to model the hierarchy of levels at which bibliographic resources can be considered and described. This hierarchy consists of four entities – Work, Expression, Manifestation and Item – and is often referred to as WEMI (see figure 3.7). For example, the physical copy of the *Iliad* that sits on my desk corresponds to an Item in the FRBR model. The edition of this copy – in this case the 2nd revised edition of Murray's critical edition published in 1999 as part of the Loeb Classical Library – constitutes a FRBR Manifestation, which is exemplified by any copy (i.e. Item) of the same edition. At a

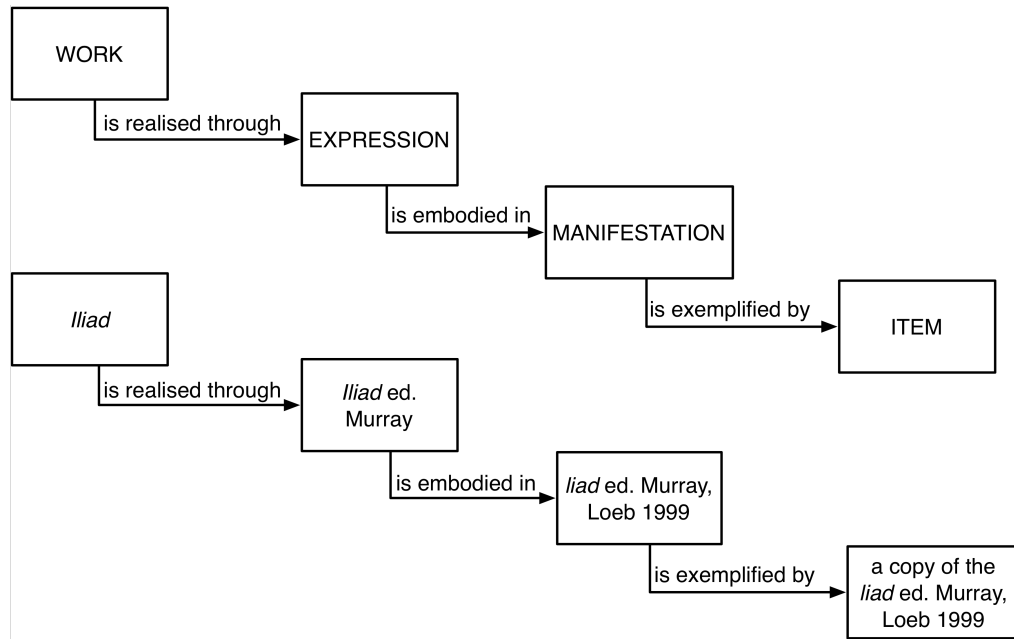


Figure 3.7: The FRBR hierarchy: an example applied to modelling the bibliographic description of a copy of the *Iliad*.

more abstract level, Murray's edition of the *Iliad* can be considered as an Expression. In this sense, the 1924 and 1999 editions of his (critical) edition are two Manifestations of the same Expression or, using the FRBR terminology, are different *embodiments* of that Expression. Finally, at the highest level of abstraction, what all editions of the *Iliad* have in common is their abstract content – or Work – of which they are considered to be realisations.

FRBR and the Canonical Text Services Protocol

Scaife (2006) first noticed a problem that is often faced when building digital libraries. The problem is that sometimes single works get represented as two or more different records. For example, the TLG Canon has no unique identifier for the Lexicon of Hesychius. Instead, the Lexicon is assigned the identifiers 4085.002 and 4085.003 as its text is drawn from two different editions – Latte and Schmidt. FRBR provides a solution to this problem as it allows us to distinguish between Hesychius's *Lexicon* as a Work and its different editions or Expressions.

Moreover, Scaife's observation highlighted two distinct yet intertwined issues: first, the need for an electronic catalogue of classical texts based on FRBR and, second, the need for a set of identifiers addressing the various levels of the FRBR hierarchy. The two initiatives that set out to

tackle these two issues – the Perseus FRBR catalogue and the Canonical Text Services (CTS) – now form the backbone of the current digital infrastructure for research in Classics, what Crane et al. (2009) called *cyberinfrastructure*.²⁰

FRBR_{OO} and CIDOC CRM

The FRBR model presented so far is also called FRBR Entity-Relationship (FRBR_{ER}) to distinguish it from another implementation of the same model, i.e. the FRBR_{OO}. The two implementations differ in the underlying methodology: the former was developed using an entity-relationship model, which is typically used to design relational databases, while the latter follows an objected-oriented model, which defines hierarchies of objects as well as the relations existing between them. The development of FRBR_{OO} was the result of harmonising the FRBR model with the CIDOC CRM, a high-level ontology for the cultural heritage sector created by the International Committee for Documentation (CIDOC) of the International Council of Museums (ICOM) (Doerr and LeBoeuf, 2007; Le Boeuf, 2012).²¹

Before discussing the implications of the aforementioned harmonisation process, let us consider briefly the goals and main principles of the CIDOC CRM. It is a very general conceptual model for expressing museum documentation with the goal of facilitating information exchange or, in more technical terms, semantic interoperability (Doerr and Iorizzo, 2008). One characteristic of information in the cultural heritage domain is that it may consist of contradictory (or even nonsensical) observations. Two records of the same museum object, for example, may assert that the same object has different dimensions; this may be due to measurements performed at different points in time or with different instruments. In order to allow for expressing this kind of information, the CIDOC CRM is based on the idea of an event. In this case the dimensions of an object are the result of a measurement event during which someone measured that object at a given place and time. Or in other words, an event is a temporal entity that occurs at a given place

²⁰ Different stages in the creation of the Perseus Catalog are described by David Mimno (2005), Babeu (2008), Crane et al. (2014) and Almas et al. (2014). Initial work on the CTS protocol by Porter et al. (2006) has been carried on by Smith (2009,0). The CTS is discussed in more details in section 3.4.3.

²¹ On the CIDOC CRM see *infra* at p. 62.

and within a certain time span; actors may participate in the event and physical as well as conceptual entities may affect it.

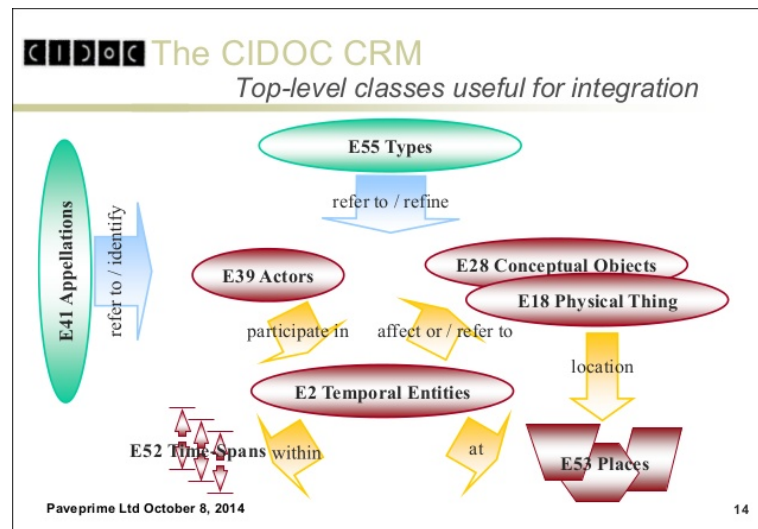


Figure 3.8: The key entities of the CIDOC CRM.

In addition to events, two other concepts play a key role in the model: appellations and types (figure 3.8). Appellations are the names that are used to refer to things; modelling appellations is key to the resolution of co-reference, i.e. the fact that two names refer to the same *thing* (e.g. “Erodoto” and “Herodotus” are names in different languages of the same author). Types allow us to express categorisations and classification systems, whose creation is part of the documentation process itself.

The significance of the harmonisation between FRBR and CIDOC CRM is that the same conceptual framework can be used to model information in the domain of archives and libraries as it can for the domain of museum, thus enabling interoperability across cultural heritage institutions.

The main implication of this process for the FRBR model was that its entities – i.e. Work, Expression, Manifestation and Item – had to be aligned with the CIDOC CRM entities and adapted to fit an event-based model. For example, the entity Work of FRBR_{ER}, which represents the abstract content of a bibliographic object (e.g. a book), becomes a subclass of a conceptual object in FRBR_{OO}. The creation of a Work coincides with the event of its conception in the author’s mind.²²

²² Another interesting example of the adaptations required to adapt FRBR_{ER} to the CIDOC CRM is provided by the modelling of the entity Manifestation, see Le Boeuf (2012, pp. 428–9) and Le Boeuf (2012).

3.4.2 *Citations in Formal Ontologies*

Beyond the field of library science, the use of ontologies has also been explored in the area of scholarly publishing to devise a conceptual model of bibliographic information, particularly of bibliographic citations. Before discussing in detail how citations are modelled in these ontologies, I shall first put their development into the context of recent developments in the field of semantic publishing.

Citation Ontologies and Semantic Publishing

Semantic publishing is the activity of “providing machine-readable metadata for journal publications and other data sources, using agreed semantic web standards that permit computers to assist in the tasks of information discovery and integration” (Shotton, 2009, p. 86). From the outset, bibliographic citations were identified as a kind of metadata that lends itself well to being semantically enriched. Once citations have been explicitly and semantically encoded, it becomes possible to explore the web of cited publications, thus allowing – among other things – for automated information discovery and gathering.

One field that has pioneered and led research in the area of semantic publishing is biomedicine. Given the sheer amount of publications in this area, the benefit of using semantic technologies to enable readers to find information more effectively has been recognised. Semantic Web Applications in Neuromedicine (SWAN) is an ontology that aims to formalise the various elements of biomedical scientific discourse (Ciccarese et al., 2008). At its core, the ontology defines discourse elements such as research statements and research questions. Bibliographic information, and particularly citations, play a key role in SWAN as they constitute the link between such discourse elements and their bibliographic sources. To this end, the ontology provides the property `swan:citesAsEvidence` that enables the explicit encoding of the publication that is evidence for a given element of the scientific discourse.

Another ontology originating from the biomedical domain is the Citation Typing Ontology (CiTO), which allows for further characterisations of the intentions – be it explicit or implicit – that lie behind the act of citing (Shotton et al., 2009; Shotton, 2010). The core of the ontology is constituted by the property `cito:cites` which represents the relation between the citing and the cited document. CiTO defines sub-

classes of this property in order to express more precisely the reasons underlying the citation. Sub-properties that derive from `cito:cites` are, for example, `cito:cites_as_evidence`, `cito:disagrees_with`, and `cito:plagiarizes`.

In the last couple of years CiTO has undergone a process of modularisation. As part of this process some concepts have been moved out of the ontology to form the Semantic Publishing and Referencing (SPAR) family of ontologies. Moreover, CiTO and SWAN have been recently harmonised due to the overlap between the two ontologies determined by the characterisation of citations (Ciccarese et al., 2014).

The potential that lies in the analysis of citation data structured using these ontologies has been recently demonstrated by the JISC-funded Open Citations project (Silvio Peroni et al., 2015). This project implemented a workflow to import citation data automatically from the PubMed biomedical archive and represent it using the SPAR ontologies. The resulting database allows users to explore, analyse and visualise the networks of citations between the articles in PubMed.

Among the ontologies included in SPAR, the FRBR-aligned Bibliographic Ontology (FaBiO) ontology is particularly relevant in the context of this chapter (Peroni and Shotton, 2012). FaBiO uses the FRBR model as a basis to represent the bibliographic records of publications. This enables us to model the fact that a research paper may have multiple realisations such as a conference paper and a journal article. These realisations are FRBR Expressions and the abstract content of the research paper is considered as the Work. Moreover, using FaBiO it is possible to link together multiple formats of the same publication – such as the electronic and the printed version of an article – as they are treated as Manifestations of the same Expression.

I turn now to examine the two main approaches to modelling bibliographic citations that can be identified in the literature. The first stresses the performative aspect of a citation, that is, it considers it as the act of linking two documents by means of a scholarly relationship. The second focuses more on the textual dimension of a citation as a series of symbols that can be interpreted as a pointer to another bibliographic entity. It is important to underline that the two approaches are not antithetical, in fact both often coexist within a single data model or ontology. Both approaches will now be considered in more detail.

Citations as Performative Entities

Citations as performative entities can be encoded simply as a relation between two bibliographical objects, or indirectly via a reified entity that represents the ‘reference event’ – the act of referencing another object. I have found no evidence of the latter but many instances of the former. For example the Bibliographic Ontology (BIBO)²³ uses the `bibo:cites` object property that relates “one document to another document that is cited by the first document as a reference, comment, review, quotation or for another purpose”. In the AKT Reference Ontology²⁴ the `akt:cites-publication-reference` is used very similarly.

More interestingly, ontologies like SWAN and CiTO, which consider citations as performative entities, go one step further by providing an explicit categorization of the types of rhetorical actions the citing relationship could indicate. For example, by using CiTO one can specify whether a citation provides evidence for some statement (e.g. `cito:cites-as-evidence`) or is aimed at criticising some ideas or statements contained in the cited work (e.g. `cito:critiques`), thus allowing the creation of networks of scholarly bibliographical relationships that are semantically very rich.

Citations as Textual Entities

In the second approach to modelling citations the focus is on the specific form a citation takes in the main body of a paper or within the references section (for this reason, it is often called reference). In other words, the emphasis here is on the symbolic level of a citation: what text it contains, how it is structured or how it is ordered. A citation object, thus intended, plays the same role as an address: it gives you useful information for finding an article. So for example in the Bibliographic Reference Ontology (BiRO)²⁵ a `biro:BibliographicReference` is seen as a textual component, which is normally part of a `biro:ReferenceList`. A similar approach is taken in the Document Components Ontology (DoCO)²⁶ and Discourse Elements Ontology (DEO)²⁷ where a `doco:BibliographicReferenceList` is said to contain one or more `deo:BibliographicReference`.

²³ Bibliographic ontology specification, <http://bibliontology.com/>.

²⁴ AKT reference ontology, <http://www.aktors.org/ontology.htm>.

²⁵ BiRO, the Bibliographic Reference Ontology, <http://vocab.ox.ac.uk/biro>.

²⁶ DoCO, the Document Components Ontology, <http://purl.org/spar/doco>.

²⁷ The Discourse Elements Ontology, <http://purl.org/spar/deo>.

Arguably, the most comprehensive formalisation of this idea is the one provided by Gruber in his Bibliographic-Data ontology²⁸. It predates the era of LOD as in fact it was not implemented using any of the RDF family of languages but does contain a number of useful and valuable insights. In particular, Gruber elucidates the terminological ambiguity between citation and reference by observing that a “reference is distinguished from a citation, which occurs in the body of a document and points to a reference” and that a “bibliographic reference is a description of some publication that uniquely identifies it, providing the information needed to retrieve the associated document”. Based on these central ideas he provides a detailed characterization of subclasses of Publication-Reference such as the development of reference-formatting styles that are independent of database or tool.

3.4.3 *Citations in Digital Classics*

Although no formal ontologies of canonical citations exist to date – a gap that HuCit aims to fill – considerable efforts have been made in the field of Digital Classics to develop standards to transform such citations into machine-actionable links. Such standards, which are reviewed in this section, are important in the context of my work insofar as they help us identify the essential properties of canonical citations. I conclude by reporting how recent activities aimed at increasing the interconnectivity between digital resources in this field – the Linked Ancient World Data (LAWD) – have facilitated the convergence between these separate standards and initiatives.

The Classical World Knowledge Base

The first standard to be considered is the Key/Encoded-Value Metadata Format for Canonical Citations²⁹, based on the OpenURL standard³⁰.

The motivation behind the development of this standard was to link the canonical citations contained in the L'Année Philologique (APh) to online resources containing the cited passage. Such a standard provides the technical foundation for the Classical Works Knowledge Base

²⁸ The Bibliographic-Data Ontology, <http://www-ksl.stanford.edu/knowledge-sharing/ontologies/html/bibliographic-data/bibliographic-data.lisp.html>.

²⁹ KEV Format: canonical citation, http://www.openurl.info/registry/docs/mtx/info:ofi/fmt:kev:mtx:canonical_cit.

³⁰ Canonical Citation Metadata Format, <http://cwkb.org/matrix/20100922/>.

(CWKB), a service of the American Philological Association (APA), aiming to provide unified access to several digital libraries of Greek and Latin texts.

The CWKB service consists of two components: first, a standard format describing how citation information can be encoded within a Uniform Resource Locator (URL) as a list of key/value pairs; second, a database of author names and work titles that enables link resolution, i.e. returning to the user a set of links pointing to online resources containing the cited passage.

Consider the following example of an OpenURL corresponding to the citation Hom. *Il.* 1.1–10:

```
http://cwkb.org/resolver?rft.au=Homer&rft.title=Iliad
&rft.slevel1=1&rft.elevel1=1&rft.slevel2=1&rft.elevel2=10
&rft_val_fmt=info:ofi/fmt:kev:mtx:canonical_cit &ctx_
ver=Z39.88-2004
```

Passing this link to the CWKB resolution service returns a human-readable list of links to editions and translations of the cited passage available online (figure 3.9).

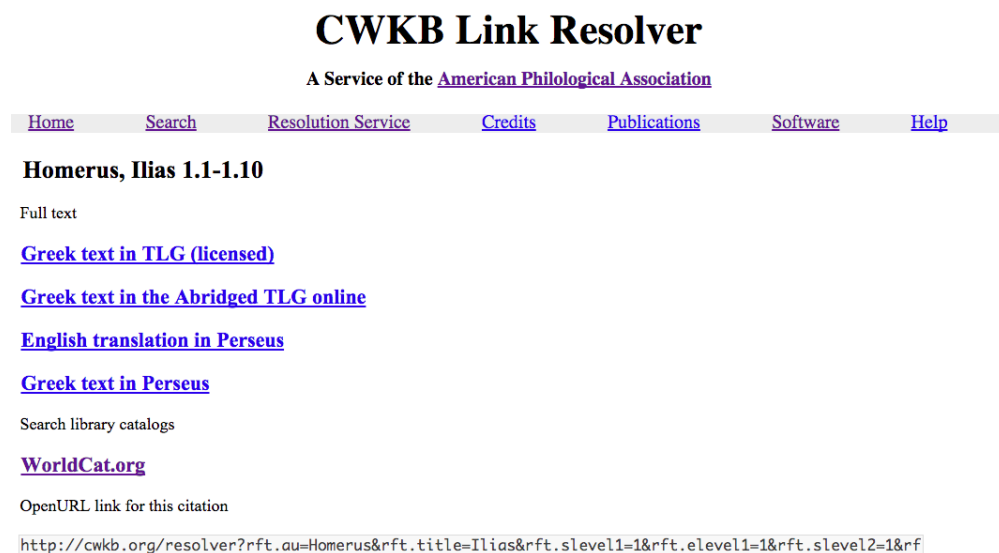


Figure 3.9: Screenshot of an example response from the CWKB resolution service.

This URL contains the following key/value pairs, separated by ampersand symbols:

- ctx_ver=Z39.88-2004: identifies the link as being an OpenURL;

- `&rft_val_fmt=info:ofi/fmt:kev:mtx:canonical_cit`: identifies the metadata format;
- `rft.au=Homer`: identifies the cited author;
- `rft.title=Iliad`: identifies the cited work;
- `rft.slevel1=1`: identifies the first hierarchical level where the citation starts (i.e. book 1);
- `rft.elevel1=1`: identifies the first hierarchical level where the citation ends (i.e. book 1);
- `rft.slevel2=1`: identifies the second hierarchical level where the citation starts (i.e. line 1);
- `rft.elevel2=10`: identifies the second hierarchical level where the citation ends (i.e. line 10).

Thanks to the underlying knowledge base, the resolution service can accept as input author names and work titles in a number of languages – i.e. Latin, English, French, German and Italian – or, alternatively, supports the use of author and work identifiers established by the Packard Humanities Institute (PHI) and Thesaurus Linguae Graecae (TLG). As a result, the user is not required to know the identifier for a given work in order to transform a citation into an OpenURL that can be resolved by the CWKB service.

One essential difference between this standard and the CTS protocol is that the former is conceived so as to enable (human) readers to look up canonical citations in online resources, whereas the latter focusses on ways to enable machines – i.e. software programmes – to resolve such citations.

The Canonical Text Service Protocol

The CTS protocol was developed in the framework of the Homer Multitext project³¹ and was designed to provide access to electronic texts in a way that is conceptually identical to how scholars have been working with such texts for centuries. To this end, it was necessary to replicate in a digital environment the system of canonical citations which allows

³¹ The Homer Multitext, <http://www.homermultitext.org/>.

scholars to create references to texts that are fine-grained and at the same time independent from any specific version of a text.

The main goal of the protocol is to make a repository of electronic texts accessible according to the canonical citation schemes of the texts it contains. The protocol allows for operations such as listing the contents of the repository or fetching the text corresponding to a given canonical citation. These operations are specified by the CTS Application Programming Interface (API), i.e. an interface that is specifically designed to be used by machines rather than by human users. Such an API provides access to any repository of texts that are encoded following the Text Encoding Initiative (TEI) guidelines. The only precondition is that the hierarchical structures according to which these texts are cited have been explicitly encoded using the appropriate TEI elements.

The CTS protocol draws its model of texts from FRBR, from which it differs in two respects. First, there is no CTS equivalent for the entity *Manifestation*, whereas *work*, *version* and *exemplar* correspond respectively to *Work*, *Expression* and *Item* in FRBR. In fact, the texts contained in a CTS repository can be referred to at different levels, i.e. at the level of *work*, at the level of *version* (i.e. edition or translation) and at the level of a specific *exemplar*. Second, CTS replaces the concept of *author* with one of *textgroup* as a means of grouping texts contained in a repository. By doing so, the protocol relaxes the assumption made concerning the authorship of a text. For example, the *textgroup* “Homer” groups together the texts of homeric poetry without necessarily assuming that they are to be attributed to an author called Homer.

urn:cts:greekLit:tlg0012.tlg001.perseus-grc1:1.1-1.10
<div style="display: flex; justify-content: space-around; border-top: 1px solid black; padding-top: 2px;"> prefix namespace textgroup work exemplar passage </div> <div style="display: flex; justify-content: center; margin-top: 5px;"> <div style="border-top: 1px solid black; width: 100%;"></div> work </div>

Figure 3.10: The syntax of a CTS URN.

A key component of the protocol are the unique identifiers that are used to identify, and therefore also to retrieve, portions of canonical texts, i.e. the CTS Uniform Resource Names (URNs).³² Their granularity varies as they allow one to refer to an entire book of a poem as well as to a single word in a line of that poem as found in a specific edition of the text.

³² For a detailed description of the CTS URN notation see Blackwell and Smith (2012).

A CTS URN consists of four colon-separated components as shown in figure 3.10:

- the prefix indicating that the identifier is a URN and follows the CTS syntax;
- the namespace identifying a specific set of identifiers, in this case the sub-set of Greek authors and works;
- the work identifier consisting of four dot-separated elements: two mandatory – i.e. the textgroup and the work identifier – and two optional elements – i.e. the version and exemplar identifier;
- the cited passage.

Using the same syntax, the following additional identifiers can be created to point at different hierarchical levels:

- `urn:cts:greekLit:tlg0012` identifies Homer (i.e. the textgroup `tlg0012`);
- `urn:cts:greekLit:tlg0012.tlg001` identifies Homer's *Iliad* (i.e. the work `tlg001` within textgroup `tlg0012`);
- `urn:cts:greekLit:tlg0012.tlg001:1.1-1.10` identifies book 1, lines 1–10 of Homer's *Iliad* without further specifying the edition or translation of the text;
- `urn:cts:greekLit:tlg0012.tlg001.perseus-grc1:1.1-1.10` identifies the same passage as above but in the edition `perseus-grc1`, i.e. the electronic version of the critical edition by Murray contained in the Perseus Digital Library.

The CTS protocol defines a number of methods to interact with a text repository. The examples that follow refer to the CTS interface implemented by the Perseus Digital Library, which makes it possible to query its contents by using the methods defined by the CTS.³³

³³ The CTS API of Perseus is accessible at <http://www.perseus.tufts.edu/hopper/CTS>.

```

<?xml version="1.0" encoding="UTF-8"?>
<TextInventory>
  [...]
  <textgroup projid="latinLit:phi0690"
    ↪ urn="urn:cts:latinLit:phi0690">
    <groupname xml:lang="eng">P. Vergilius Maro</groupname>
    <work projid="latinLit:phi003"
      ↪ urn="urn:cts:latinLit:phi0690.phi003" xml:lang="lat">
      <title xml:lang="eng">Aeneid</title>
      [...]
      <edition projid="latinLit:perseus-lat1"
        ↪ urn="urn:cts:latinLit:phi0690.phi003.perseus-lat1">
        <label xml:lang="eng">Aeneid</label>
        <description
          ↪ xml:lang="eng">Perseus:bib:oclc,22858571, Vergil.
          ↪ Bucolics, Aeneid, and Georgics Of Vergil. J. B.
          ↪ Greenough. Boston. Ginn & amp; Co.
          ↪ 1900.</description>
        [...]
      </edition>
    </work>
  </textgroup>
  [...]
</TextInventory>

```

Figure 3.11: The XML reply of the Perseus' CTS API upon a GetCapabilities request request (some details omitted for the sake of readability).

The method `GetCapabilities`, which takes no additional input parameters, returns an XML catalogue of the texts available in the repository. The catalogue of texts is organised according to the hierarchy of textgroups, works, editions and exemplars that was already described. Figure 3.11 shows an excerpt of the response returned by the Perseus' API upon such a request, specifically the catalogue record corresponding to the *Aeneid*.

```

<?xml version="1.0" encoding="UTF-8"?>
<cts:GetPassage>
  <cts:request>
    <cts:requestName>GetPassage</cts:requestName>
    <cts:requestUrn>urn:cts:latinLit:
      ↪ phi0690.phi003:1.1</cts:requestUrn>
    <cts:psg>1.1</cts:psg>
    <cts:workUrn>urn:cts:latinLit:phi0690.phi003</cts:workUrn>
    <cts:groupname>P. Vergilius Maro</cts:groupname>
    <cts:title>Aeneid</cts:title>
    <cts:label>Aeneid</cts:label>
    <cts:versionInfo>Perseus 4.0</cts:versionInfo>
  </cts:request>
  <cts:reply>
    <tei:TEI>
      <tei:text xml:lang="la">
        <tei:body>
          <tei:div type="line">Arma virumque cano, Troiae qui
            ↪ primus ab oris</tei:div>
        </tei:body>
      </tei:text>
    </tei:TEI>
  </cts:reply>
</cts:GetPassage>

```

Figure 3.12: The XML reply of the Perseus' CTS API upon a GetPassage request (some details omitted for the sake of readability).

The method GetPassage provides a way to look up a canonical citation. It takes as an input parameter the URN of the passage to look up and it returns an XML reply containing the text of that passage encoded as TEI. Figure 3.12 shows the reply upon a GetPassage request with the input parameter urn:cts:latinLit:phi0690.phi003:1.1, the CTS URN corresponding to Vergil, *Aen.* 1.1.

```

<?xml version="1.0" encoding="UTF-8"?>
<GetValidReff xmlns="http://chs.harvard.edu/xmlns/cts3/ti">
  <request>
    <requestName>GetValidReff</requestName>
    <requestUrn>urn:cts:latinLit:phi0690.phi003
      ↪ .perseus-lat1</requestUrn>
  </request>
  <reply>
    <reff>
      <urn>urn:cts:latinLit:phi0690.phi003.perseus-lat1:1.1</urn>
      <urn>urn:cts:latinLit:phi0690.phi003.perseus-lat1:1.2</urn>
      <urn>urn:cts:latinLit:phi0690.phi003.perseus-lat1:1.3</urn>
      <urn>urn:cts:latinLit:phi0690.phi003.perseus-lat1:1.4</urn>
      <urn>urn:cts:latinLit:phi0690.phi003.perseus-lat1:1.5</urn>
      [...]
    </reff>
  </reply>
</GetValidReff>

```

Figure 3.13: The XML reply of the Perseus' CTS API upon a GetValidReff request (some details omitted for the sake of readability).

Finally, the method `GetValidReff` returns an ordered list of citable passages for the work specified by the URN input parameter. To get a list of all the citable passages of the *Aeneid*, for example, one can call this method while passing as a parameter the identifier corresponding to the *Aeneid*, i.e. `urn:cts:latinLit:phi0690.phi003` (see figure 3.13). This method can be used for validation purposes as it allows us to verify that a given CTS URN can actually be resolved.

The Linked Ancient World Data

Before moving to the discussion of the HuCit ontology, the role played by the CTS protocol in the emerging LAWD deserves to be considered.

LAWD is a community-led effort to apply the LOD approach to digital resources that are available to study the ancient world. LAWD activities were sparked by the Linked Ancient World Data Institute (LAWDI), a series of two events founded by the National Endowment of the Humanities that took place in 2012 at New York University and in 2013 at Drew University. These institutes fostered collaboration between people and projects which was seen as the first step to increase the interconnectivity between digital resources (Elliott et al., 2014). For this reason, LAWDI

has focussed more on (the necessity of) making resources available and creating links between them than, for example, on developing shared ontologies.³⁴

Since the essence of LOD is connecting datasets by means of links, a crucial role is played by those resources that provide names – i.e. URIs – to identify places, people, museum objects, texts, etc. as they allow other datasets to link to them. The Pleiades gazetteer, for example, enabled the Pelagios project to aggregate a variety of digital resources based on their references to geographical places (Simon et al., 2014). Pelagios relies on the Pleiades URIs as a shared vocabulary to describe relations between places and resources and, by doing so, enables the creation of links between such resources.

Similarly, CTS URNs are becoming the shared set of identifiers to link together resources related to ancient texts.³⁵ LAWDI has helped to overcome the two factors that have hindered the adoption of these identifiers in a LOD context. First, the fact that CTS URNs *per se* are not resolvable identifiers, thus they cannot be used as the subject or object of an RDF triple in a LOD-compliant way. Second, the lack of a central registry of CTS URNs. In this respect Linked Ancient World Data Institute has facilitated the adoption by Perseus of the CTS architecture for both its repository of texts – the Perseus Digital Library – and its catalogue of metadata about ancient authors and works, i.e. the Perseus Catalog. In doing so, not only has Perseus become a *de facto* registry of CTS URNs, but it has also turned CTS URNs into resolvable URIs, thus making them usable in a LOD context (Almas et al., 2014).

Another activity that has spun off from LAWDI is the development of a LOD API to expose the CWKB data ³⁶. As a result, the approximately 1,550 authors and 5,200 works contained in CWKB are now identified by URIs and linked, whenever possible, to the corresponding record in the Perseus Catalog. Figure 3.14 shows the triples that are returned when resolving <http://cwkb.org/work/id/1413/turtle>, the URI for Vergil's *Eclogues*. Three details of this record are worth noting:

³⁴ Ontology development, despite being relatively marginal, led to the LAWD ontology, “a minimal ontology for connecting vocabularies useful in describing data concerning the ancient world”, available at <http://lawd.info>.

³⁵ It must be noted that, in addition to the projects discussed in this section, the CTS protocol was adopted by the project Sharing Ancient Wisdoms (SAWS) (Roueché et al., 2014) and its integration into Tesseract is already planned for the next version of the software (Coffee, 2014). See section 2.3 for further details on both projects.

³⁶ Canonical Citation Linking and OpenURL / API, <http://cwkb.org/lod>.

```

@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix lawd: <http://lawd.info/ontology/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://cwkb.org/work/id/1411/rdfa> a lawd:conceptualWork;
  rdfs:label "Bucólicas"@es,
    "Églogas"@es,
    "Bucolica"@la,
    "Bucoliche"@it,
    "Egloghe"@it,
    "Eklogen"@de,
    "Hirtengedichte"@de,
    "Bucoliques"@fr,
    "Éclogues"@fr,
    "Bucolics"@en,
    "Eclogues"@en;
  lawd:abbreviation "ecl.";
  lawd:citation "http://cwkb.org/resolver?rft.auauthority=P.
    → Vergilius Maro&rft.titleauthority=Eclogae
    &rft_val_fmt=info:ofi/fmt:kev:mtx:canonical_cit
    &ctx_ver=Z39.88-2004";
  lawd:responsibleAgent "http://cwkb.org/author/id/678/rdfa";
  dcterms:identifier "phi:0690.001",
    "urn:cts:latinLit:phi0690.phi001";
  dcterms:language "latin";
  owl:sameAs
    → <http://data.perseus.org/catalog/urn:cts:latinLit:phi0690.phi001>
    → .

```

Figure 3.14: CWKB: the record for Vergil's *Eclogues*, expressed using the Turtle syntax.

- 1) the `dcterms:identifier` property indicates the CTS URN for this text;
- 2) the `owl:sameAs` property links to the corresponding record in the Perseus Catalog;
- 3) the `lawd:citation` property provides the OpenURL for this text.

3.5 A FORMAL MODEL OF CITATIONS: THE HUCIT ONTOLOGY

In this section I present the classes and properties that form the HuCit ontology, which aims to represent canonical citations after they have been extracted from texts. The design of the ontology was informed by

the domain analysis discussed in section 3.2 and by the existing models of citations reviewed in section 3.4.

In section 3.5.1 I discuss the methodology and principles that I followed in developing the ontology. In section 3.5.2 I give an overview of the classes defined in HuCit and explain how they relate to the ontologies that HuCit extends, i.e. CIDOC CRM and FRBR_{OO}. The rest of the section is dedicated to examining the classes of the ontology and is divided into three thematic areas: the classes that model a citation and the citing document (section 3.5.3); the classes that model the content of a citation and the structure of the cited text (section 3.5.4); the classes that represent information about the cited text and its author such as abbreviations, author names and work titles (section 3.5.5).

3.5.1 *Methodology and Rationale*

There were two reasons for manually creating HuCit instead of eliciting it from a corpus of texts by means of ontology learning.

First, as observed in section 3.2, the way classicists cite literary texts already implies and reflects a model of those texts. Therefore an ontology of canonical citations can be created by analysing the citation practices and formalising them as a set of classes and relationships between those classes. Moreover, the very process of modelling is valuable and interesting in itself. In fact, modelling forces us to reflect on how we perceive what we model and, by doing so, it allows us to gain a deeper understanding of the reality modelled.

Second, the ontology learning approach turned out not to be suitable for creating an ontology of canonical citations as was shown by the results of a preliminary experiment. In this experiment a word clustering technique called *Latent Semantic Analysis* was used to extract from a corpus of 170 journal articles related to Classics a set of key terms (Romanello et al., 2009b, pp. 158–160). The extracted terms were then grouped together with their associated terms into semantic clusters. Although this experiment did lead to some interesting results, it did not provide any further evidence as to how scholars perceive canonical citations.

An interesting result obtained with this experiment was a set of concepts of importance in the domain of Classics, clustered into three groups. The first cluster consists of words related to text editing such

as “copyist”, “emendation”, “variant” and “corruption”. The second cluster contains words that describe the spatial relations between parts of a text, such as “line”, “beginning”, “end” and “margin”. Finally, the third cluster consists of words such as “evidence”, “authenticity”, “uncertainty” and “interpretation”. These words belong to the sphere of scrutiny and subjectivity and play a key role in philology and text editing.

There can be various reasons why ontology learning failed to capture terms related to canonical citations in the experiment just described. This may be due to how the sample was selected, its size or the fact that classicists *use* such citations rather than *theorise* or *speculate* about their origin or characteristics. Even when they do talk about them – such as in resources providing guidelines to students on how to properly cite ancient texts – they do so in a rather pragmatic way. Consider, for example, how *The Cambridge companion to Aristotle* explains the various ways to cite Aristotle (emphasis my own):

Different scholars prefer different abbreviations [...]; some scholars refer to books by number rather than by Greek letters, and some do not refer to books at all; different editions of Aristotle’s works have used different chapter divisions, and, again, some scholars do not refer to chapters. But **Bekker will rarely let you down**: virtually all later editions of the Greek texts print Bekker references in their margins; virtually all books and articles give Bekker numbers either in the margin or at the head of the page (Barnes, 1995, p. xxi).

One challenge I had to face while building HuCit was how to deal with the arbitrariness that is inherent in the manual creation of any ontology. The difficulty arises from the fact that there may be many possible ways of modelling a given phenomenon or domain which may all be or seem equally correct. In this respect, the ontology of representations defined by Mizoguchi (2004) provided a useful frame of reference to define what *is* ultimately a canonical reference. Representations are objects such as music, poems, algorithms – or citations – that, unlike other objects, have the characteristic of bearing some content. One key principle of Mizoguchi’s methodology is to distinguish between the *form* and the *content* of a representation. This simple principle proved to be

extremely useful when disentangling canonical citations by introducing the question of what constitutes the form or the content of a citation.

Another decision I took to make HuCit less arbitrary was to ground it in existing and increasingly popular ontologies such as CIDOC CRM and FRBR_{OO}. For this reason, the ontology defines a relatively small number of new classes and properties as it imports and extends a number of existing concepts from these two ontologies. In particular, HuCit makes use of high-level concepts defined by CIDOC CRM such as types, part-of relationships and identifiers. HuCit draws concepts from FRBR_{OO}, which is also based on CIDOC CRM, concepts describing texts and their authorships (e.g. author, title, etc.) as well as concepts representing the hierarchy of distinct levels of a bibliographic object (i.e. work, expression, manifestation and item). Finally, the CTS protocol, although not strictly speaking an ontology, provided a solid foundation for my work especially in the form of a set of identifiers – the CTS URNs – that are suitable to identify instances of classes defined by HuCit.³⁷

3.5.2 Overview of HuCit

The HuCit ontology consists of a small number of classes and properties and extends concepts defined by CIDOC CRM and FRBR_{OO} (Romanello and Pasin, 2011,0).³⁸

At the highest level of abstraction, it must be observed that HuCit deals with conceptual objects, i.e. the immaterial products of the human mind. In turn, as shown in figure 3.15, conceptual objects belong to that category of entities that have a persistent identity (E77_Persistent_Item) – the so-called *endurants* in philosophy – whose nature can be either physical (E24) or conceptual (E28). These entities form the top-level of CIDOC CRM along with, among others, place (E53_Place) and time (E2_Temporal_Entity).

All entities described by HuCit are conceptual objects with the only exception being the class E21_Person and its FRBR_{OO} equivalent F10_

³⁷ A similar approach based on CIDOC CRM, FRBR_{OO} and CTS was adopted to devise an ontology of relationships between or within texts for the SAWS project as described by Jordanous et al. (2012a). On SAWS see *infra* at p. 44.

³⁸ The ontology, implemented using OWL, is available online at the permanent URI <http://purl.org/net/hucit>. HuCit relies on the OWL implementations of CIDOC CRM version 5.2.1 and FRBR_{OO} version 1.0.2, originally developed by a research group at the university of Erlangen-Nürnberg and now available to the community at <https://github.com/erlangen-crm/ecrm>.

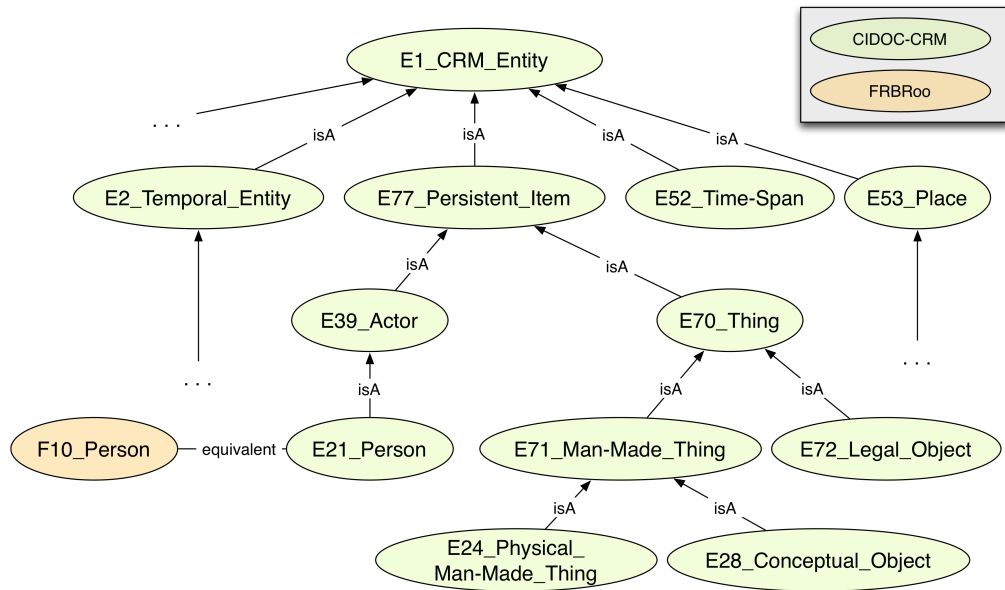


Figure 3.15: Diagram situating the class E28_Conceptual_Object within the CIDOC CRM taxonomy.

Person. This class is used in the ontology and in the knowledge base to represent the author of a given work (e.g. Aristophanes, the author of *The Clouds*).

In the rest of this section I discuss in detail the classes that constitute HuCit, their relation to CIDOC CRM and FRBR_{OO} and the modelling choices that led to their identification. The classes are divided for convenience into the following groups:

- classes to describe the document containing a canonical citation (section 3.5.3);
- classes to describe the canonical structure of the cited text (section 3.5.4);
- classes to describe authority data about classical texts such as names, titles, abbreviations (section 3.5.5).

The example abstract drawn from the APh and shown in figure 3.16 is used in the following sections to illustrate the usage of HuCit to model and express the meaning of canonical references.

3.5.3 Modelling a Citation and the Citing Document

HuCIt defines four classes to describe the content of the documents from which canonical references are extracted – the abstracts of the

Text of APh 75-06176 (canonical references highlighted in bold)

Vergil has been appropriated for many different perspectives on the world. These appropriations involve assimilations that tend to erase the particularly Roman aspects of **Vergil**. It is important to make, or keep, **Vergil** strange, especially in the area of translation. A defamiliarizing reading of **Aen. 1,1-11** helps to illustrate how some English translations of the « **Aeneid** » map onto the spectrum of assimilation-dissimilation.

Figure 3.16: An example abstract drawn from the APh.

APh and the journal articles in JSTOR. These classes – i.e. Document, Sentence, Citation and CanonicalCitation – are needed to represent the hierarchical structure of such documents. Each document is made of sentences which, in turn, may contain citations. This simple and relatively flat structure also reflects the format of documents as they are processed by the citation extraction system described in chapter 4.

Figure 3.17 shows how these classes are situated within and connected to the class hierarchy of CIDOC CRM and FRBR_{OO}. Documents and their content are modelled at an abstract level and are treated as conceptual objects having both a form and content (i.e. information objects), where the form is the set of signs used to express their abstract content. HuCit deliberately does not deal with the bibliographic description of such documents as this is dealt with other ontologies such as the aforementioned FaBiO. In fact, since the RDF model permits to say that an entity is simultaneously an instance of multiple classes, the example document APh 75-06176 could be declared as an instance of both Document and fabio:PeriodicalItem, thus stating that the document is a part of a given volume of the periodical publication *L'Année Philologique*.

The classes Document, Sentence and Citation correspond to the level of Expression in the FRBR hierarchy and are thus defined as subclasses of F2_Expression. Moreover, since the form of these documents is text, they are simultaneously also subclasses of E33_Linguistic_Object, which allows us to record the language in which they are written via their property P72_has_language. It is also worth noting that although only one subclass of Citation is currently defined – i.e. CanonicalCitation – more subclasses could be introduced in the future to describe other kinds of citations that are found throughout publications in

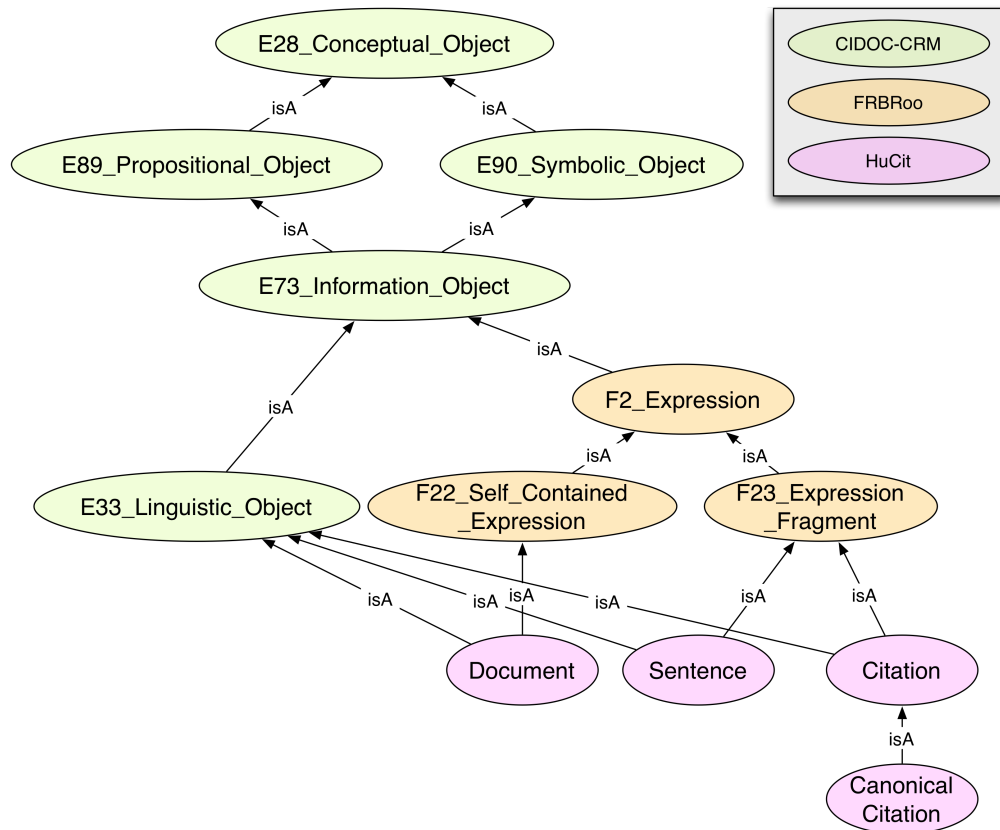


Figure 3.17: HuCit ontology: classes Document, Sentence, Citation and CanonicalCitation and their relation to CIDOC CRM and FRBR_{OO}.

Classics (e.g. citations of inscriptions, papyri, manuscripts fragmentary texts, etc.).

Figure 3.18 shows how the classes introduced thus far can be used to model the content of our example.

I turn now to consider in detail how canonical citations were modelled. Firstly, citations were modelled as a class rather than as a relation between class instances – the two approaches discussed in section 3.4.2. In fact, only the former approach allows us to retain the string of characters that constitutes a citation (in addition to its content), which is an essential feature as the ontology is to support the automatic extraction of such references.

Secondly, following Mizoguchi’s methodology to modelling the semantics of content-bearing objects, it was possible to tease out what constitutes the *form* and the *content* of a canonical reference. The form is the style used to express that reference, while the content is what is being referenced – whatever its ontological status may be. In the example citation Verg. *Aen.* 1,1–11 the form is the citation style this reference

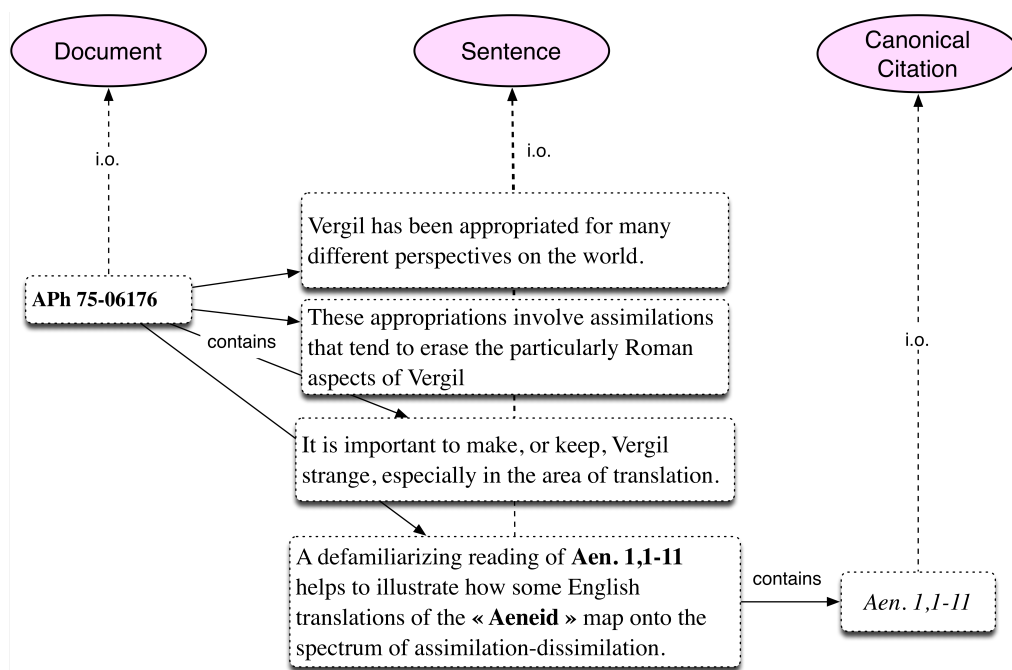


Figure 3.18: HuCit ontology: usage of classes Document, Sentence and CanonicalCitation to model the content of an APh abstract.

follows, which determines its appearance, while the content consists of the lines 1–11 of Vergil’s *Aeneid* (figure 3.19). At this point we can say these lines are instances of the class `TextElement`. A rationale for this class is discussed in detail in the next section.

Distinguishing between the form and the content of a canonical citation allows us also to determine its identity. In fact, this distinction enables us to determine formally when two citations are equivalent or identical. Two or more citations are equivalent when they express the same content but have different forms, whereas they are identical when they share both form and content. For example, the references Vergil, *Aeneid* I 1–11, Verg. *Aen.* 1,1–11 and Verg. *Aen.* 1.1–11 are all equivalent to each other but not identical. In fact, what they all mean is “book 1, lines 1–11 of Vergil’s *Aeneid*”, yet they differ as to how this meaning is expressed, in other words they follow different citation styles.

Although citation styles are defined in HuCit, I did not classify the styles of all citations that were extracted from the APh and JSTOR as this falls beyond the scope of my research. It would be possible and certainly useful, however, to define a taxonomy of citation styles for canonical references. With such a taxonomy at hand, a tool could be

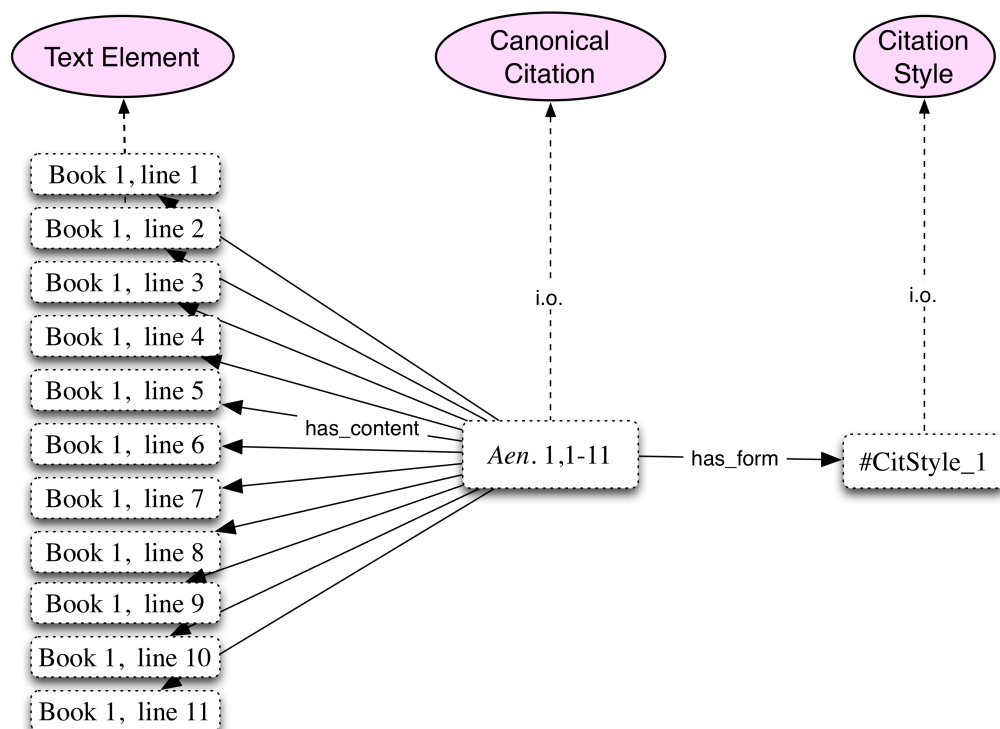


Figure 3.19: HuCit ontology: usage of classes `TextElement`, `CanonicalCitation` and `CitationStyle` to model the form and content of the citation *Aen. 1,1-11*.

implemented to automatically format the canonical citations found in a given document according to different citation styles.³⁹

3.5.4 Modelling the Structure of the Cited Text

In the previous section the content of the example citation *Aen. 1,1-11* was described as being a reference to book 1, lines 1-11 of Vergil's *Aeneid*. Let us now define what these books and lines are in more precise ontological terms.

A canonical citation does not refer *directly* to the text that corresponds to the cited passage. Instead, it refers to an abstract structure of the text – or canonical citation scheme – that corresponds to the division of an idealised edition of the text. In our example, the structure consists of books which are then divided into lines. The purpose of this structure

³⁹ A similar functionality is already provided for by modern bibliographic reference software such as Zotero <https://www.zotero.org/> or Mendeley <http://www.mendeley.com/> as they allow users to format a set of (modern) bibliographic references according to several citation styles.

is to provide the coordinates for locating the cited passage within the hierarchy of the text.

To model such text divisions HuCit provides the classes `TextStructure` and `TextElement`. The class `TextStructure` represents the hierarchical organisation of a text into more granular units such as the division of a text into books, chapters and sections or the organisation as determined by the pagination. The elements of such a hierarchical structure are instances of the class `TextElement`. Their order and the hierarchical relations between them are captured by the properties `follows` (and its inverse `precedes`) and `is_part_of` (and its inverse property `has_part`).

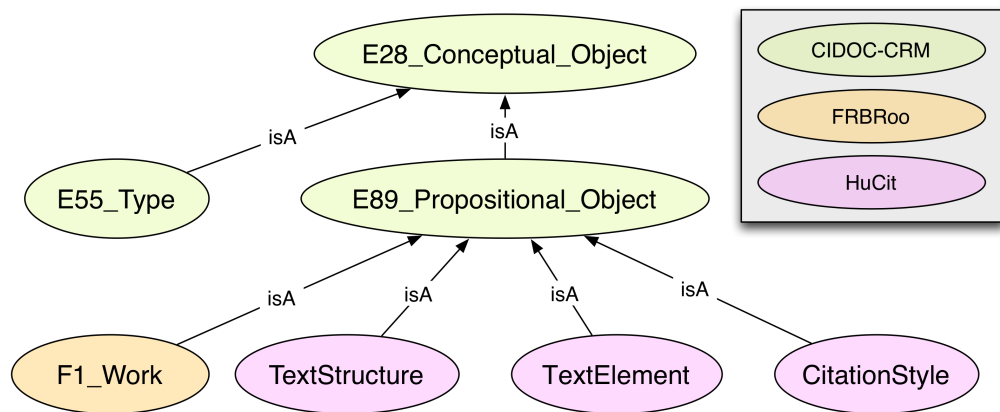


Figure 3.20: HuCit ontology: the classes used to model the structure of the cited text – `E55_Type`, `F1_Work`, `TextStructure` and `TextElement` – and their relation to `E28_Conceptual_Object`.

The classes `TextStructure` and `TextElement` represent conceptual objects and are defined as subclasses of `E89_Propositional_Objects` (see figure 3.20). In CIDOC CRM propositional objects are a specific kind of conceptual object that represents propositions – i.e. statements or assertions – *about* real or imaginary things. This class was chosen as the superclass of `TextStructure` and `TextElement` because instances of these classes can be considered as assertions made *about* texts (e.g. the text structure of Vergil’s *Aeneid*).

The aforementioned classes can now be used to model the citation contained in the example (see figure 3.21). The content of the CanonicalCitation *Aen. 1,1–11* is constituted by a set of instances of the class `TextElement` representing lines 1–11 of the first book of the *Aeneid*. The hierarchical structure of this text, which consists of books and lines, is expressed by means of the property `part_of` (e.g. book 1, line 1 is part

of book 1, etc.). The boolean property `is_canonical`, which has the value `True`, indicates that the instance of `TextStructure` is currently the canonical structure used to refer to the *Aeneid*. This property allows us to distinguish the text structures that are canonical from those that are non-canonical at a given point in time. Finally, it is worth highlighting that the property `is_structure_of` connects the instance of `TextStructure` to the instance of `F1_Work` representing in the FRBR hierarchy the abstract notion of the *Aeneid*. This modelling reflects the fact that a canonical text structure is common to virtually all editions of a text or, to use the FRBR terminology, to all expressions of a work.

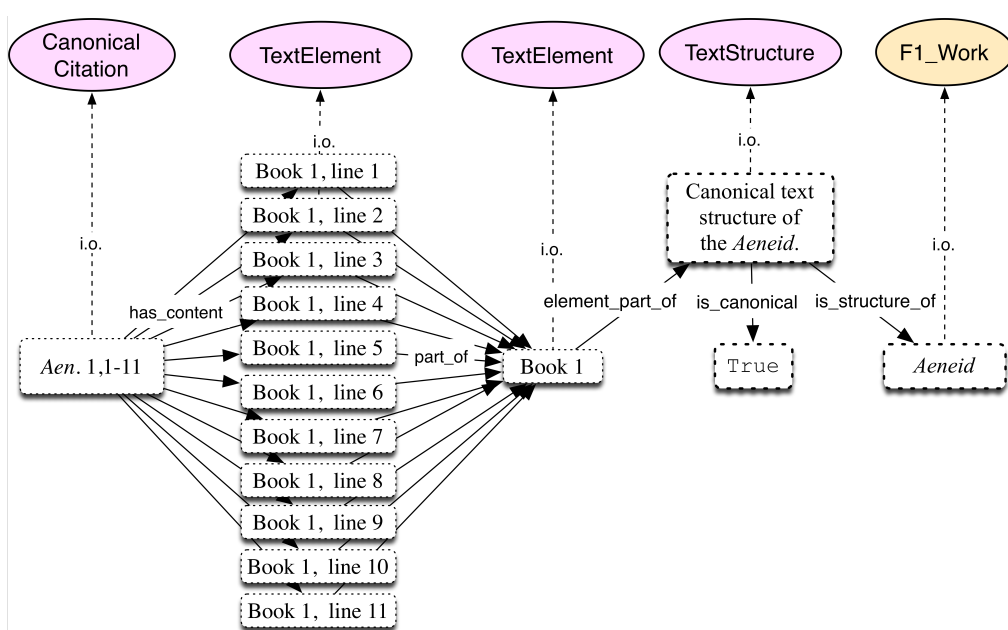


Figure 3.21: HuCit ontology: usage of classes `TextElement`, `TextStructure` and `F1_Work` to model the content of a canonical citation.

The rationale for the class `TextStructure` is worth discussing in more detail as its modelling proved to be particularly challenging and it evolved over time. The first controversial decision concerns the name of the class. In earlier versions of HuCit this class was named `CitationScheme` to reflect the terminology more commonly used when talking about canonical citations as seen in section 3.2. The class, however, was renamed to `TextStructure` based on the observation that what constitutes the structure – or scheme – for a citation is, in fact, the implied hierarchical structure of the cited text. In other words, being a citation scheme is a *role* played by a text structure rather than a class in its own right.

The second decision concerned how to express the fact that a text structure becomes the canonical way of citing that text at a given point in time. As discussed in section 3.2.2, this is the case with the pagination of the 1578 edition of Plato by Stephanus, which at some point becomes the text structure of reference to cite works by Plato. An alternative solution to specifying the property `is_canonical` would have been to introduce a subclass of `TextStructure` called `CanonicalTextStructure`. However, the problem with this solution is that in order to model the fact that this text structure becomes canonical it would be necessary to delete the existing instance of `TextStructure` and replace it with a new instance of `CanonicalTextStructure`. This issue suggested that introducing the property `is_canonical` of the class `TextStructure` was a more appropriate modelling strategy than adding a new subclass.

Furthermore, the class `TextElement` has a particular importance in HuCit as it connects the ontology with the CTS protocol. The CTS identifier corresponding to *Aen.* 1,1–11 is `urn:cts:latinLit:phi0690.phi003:1.1-1.11`. The protocol, however, does not define the ontological status of the entity indicated by this identifier, instead this is done instead by HuCit. The CTS URN above identifies a set of elements within a canonical text structure (i.e. instances of `TextElement`). As shown in figure 3.22, CTS URNs are modelled in the ontology as identifiers (`E42_Identifier`) of a given type (`E55_Type`) that are assigned to class instances.

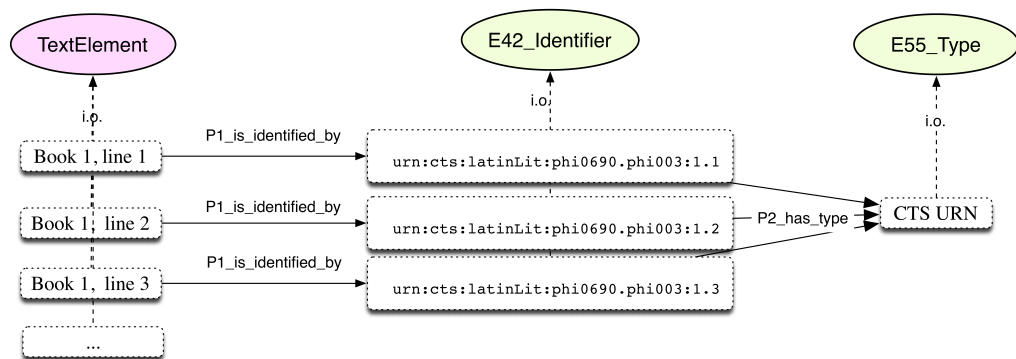


Figure 3.22: HuCit ontology: assignment of a CTS URN to the corresponding `TextElement` instance with classes `E42_Identifier` and `E55_Type`.

A final aspect to consider is the use of the class `E55_Type` defined by the CIDOC CRM to group together `TextElement` instances into a taxonomy of types. For example, the structure of texts that have books and lines as elements (e.g. Virgil's *Aeneid*, Homer's *Iliad*, Apollonius

Rhodius' *Argonautica*, etc.) are characterised by the same text element types. In light of the intended use of the knowledge base – i.e. supporting the automatic extraction of canonical citations – such a taxonomy allows labels to be defined in several languages for the same element type. This information can then be leveraged when disambiguating citations as described in section 3.6.3.

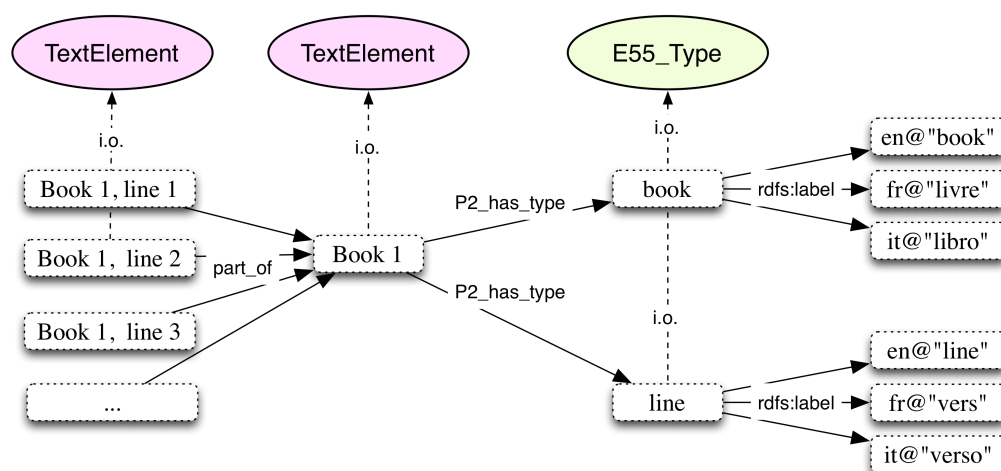


Figure 3.23: HuCit ontology: usage of class E55_Type to define a typology of TextElement instances. Such a typology enables us to classify elements of a text structure and to retain, for each type, the corresponding labels in several languages.

3.5.5 Modelling Authority Data

The last group of classes to be considered are those employed to represent authority information about ancient authors and works. This kind of information includes names of authors and titles of works in different languages together with their abbreviations and unique identifiers. Authority information plays a key role in the context of the automatic extraction and disambiguation of canonical references. In fact, it is obvious for the trained reader that the abbreviation “*Aen.*” in the reference *Aen.* 1,1–11 stands for the *Aeneid*, whereas the automatic citation extraction system needs to access this information in order to correctly capture the meaning of the canonical reference.

HuCit does not define its own classes to represent authority information. Instead, it uses the classes E35_Title, E41_Appellation and E42_Identifier provided by CIDOC CRM and FRBR_{OO} (see figure 3.24). The class E41_Appellation – and its FRBR_{OO} equivalent F12_Name – de-

scribes sets of signs that are used to refer to and identify instances of some class in a given context (e.g. the name “Vergil” to refer to the Latin author who wrote the *Aeneid*). A specific kind of appellation is the identifier (E42) – and its equivalent F13_Identifier – these are names for entities that aim to be unique and permanent within certain contexts (e.g. an ISBN code, a CTS URN, etc.). Moreover, titles (E35) are a specific kind of appellation that consist of linguistic expressions (E33) used to refer to texts, artworks etc.

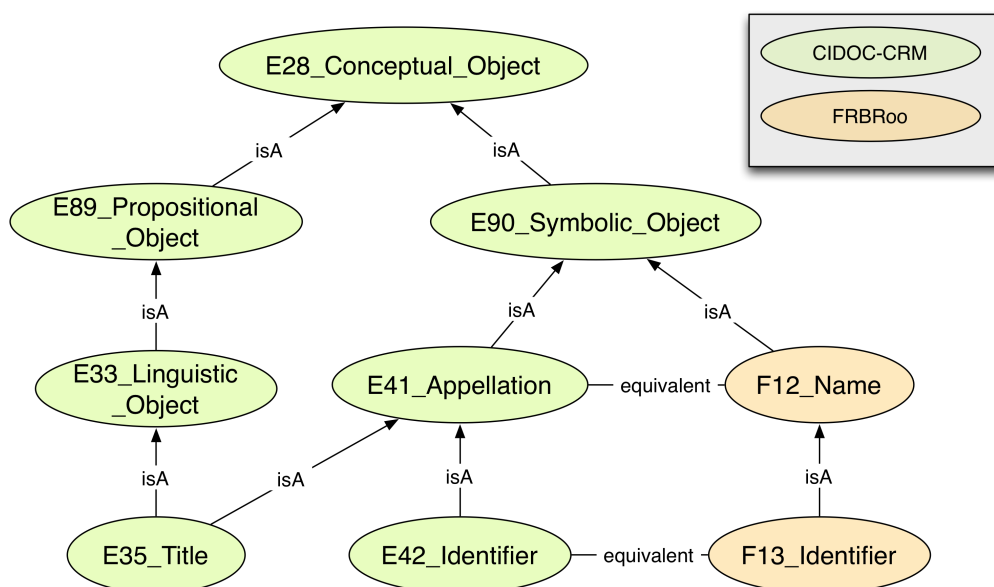


Figure 3.24: HuCit ontology: diagram situating the classes used to model authority data within the hierarchy of CIDOC CRM.

Figure 3.25 shows how these classes are employed to store authority information about Vergil, the cited author in our example. The fact that a given string is an abbreviation is expressed by associating the type abbreviation (E55) to an instance of E41_Appellation via the property P2_has_type. As shown in figure 3.25, “Verg.” is defined as an instance of appellation (E41) qualified by the type abbreviation. The property P139_has_alternative_form is then used to state that the abbreviation “Verg.” is an equivalent way of referring to the author Vergil. Moreover, the modelling of the CTS URN associated to Vergil follows the same pattern discussed in the previous section in relation to the identification of TextElement instances.

It is also worth noting that this model allows us to create a typology of abbreviations which is useful when tracking their provenance. Such a typology enables us to group together all abbreviations defined in re-

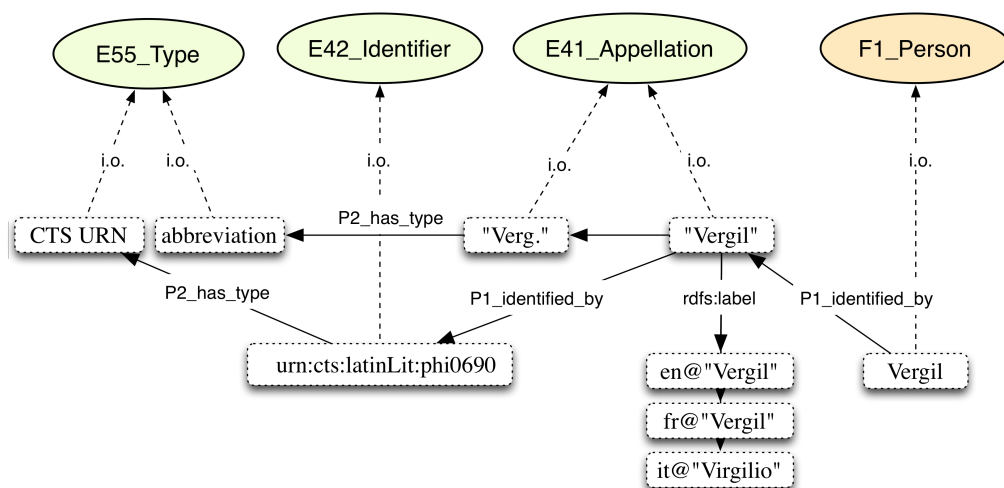


Figure 3.25: HuCit ontology: modelling the authority data related to Vergil – i.e. CTS URN, name variants and abbreviations – with classes from CIDOC CRM.

sources such as the Liddell-Scott-Jones or those used by the APh. Since the set of abbreviations used is arguably the most distinctive feature of a specific style of citing classical texts, this typology could be one of the core components of a tool developed for the automatic formatting of canonical references according to various citation styles.

In addition to variants and abbreviations of names and titles, authorship is another essential aspect of authority information that HuCit deals with. In FRBR_{OO} the authorship – i.e. the attribution of a work to its author – is modelled as an event performed by the author and leading to the creation of a given work (e.g. the *Aeneid*). The reason for such an event-based model of authorship lies in the harmonisation process of FRBR with the event-based model of CIDOC CRM that resulted in the FRBR_{OO} ontology.⁴⁰ As can be seen in figure 3.26, Vergil and the *Aeneid* are connected by the event of the conception of the poem by its author – i.e. the instance of F27_Work_Conception.

Finally, a dedicated type – represented by an instance of E55_Type – marks a work that is considered the author's opus maximum (e.g. Martial's *Epigrammata*). This piece of information is essential when automatically determining which text a citation like "Martial 1, 60, 3–4" is referring to. Since the *Epigrammata* is Martial's opus maximum, it is perfectly acceptable to leave out the indication of the work. A computer program, however, cannot infer what is actually meant by the string "Martial" unless additional information concerning the author's opus

⁴⁰ On this harmonisation process see *infra* p. 71.

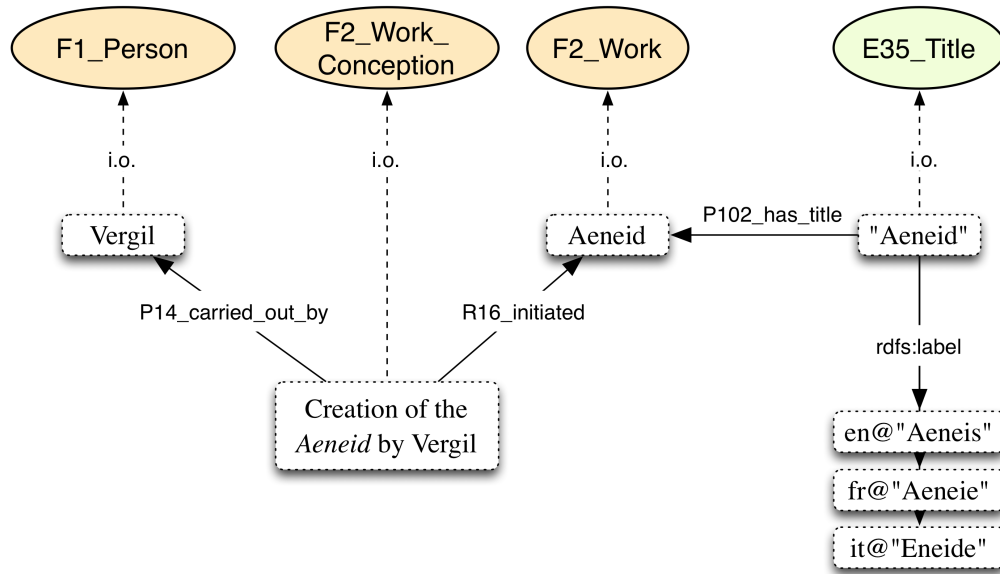


Figure 3.26: HuCit ontology: event-based modelling of the authorship of the *Aeneid* in FRBR_{OO}.

maximum is supplied by an external source such as in this case the knowledge base.

3.6 A KNOWLEDGE BASE TO SUPPORT THE EXTRACTION OF CITATIONS

In this section I first describe the implementation of a knowledge base containing instances of the ontology classes discussed in the previous section (section 3.6.1). The goal of this knowledge base is to support the automatic extraction of canonical citations. I then explain how the knowledge base was populated by leveraging already existing resources (section 3.6.2). I conclude by considering ways, both current and future, of harnessing the information of the knowledge base to improve the automatic extraction of citations (section 3.6.3).

3.6.1 Technical Implementation

The knowledge base contains instances of classes defined by HuCit, CIDOC CRM and FRBR_{OO} as well as relations between them. Its content is structured according to the LOD principles illustrated in section 3.3.3. This means that each resource in the knowledge base – e.g. authors, works, etc. – is identified by an URI which, when looked up,

returns an RDF description of the resource. The RDF triples expressing these instances and relations are served by a triple store, a database system specifically designed and optimised for storing and querying RDF data.⁴¹

The triple store I chose to implement the knowledge base is AllegroGraph⁴². Despite being a commercial graph database, it is available also as a free edition with storage capabilities limited to 5 million triples. Although several triple store solutions currently exist, both free and commercial,⁴³ I have adopted AllegroGraph as it integrates with SuRF better than other solutions that were tested. SuRF⁴⁴ is the Python library I have been using to connect the knowledge base with the code – also written in Python – that handles the extraction and disambiguation of canonical references. Another advantage of this solution is the possibility of using Gruff⁴⁵, a tool to visually inspect the content of an AllegroGraph triple store.

3.6.2 *Populating the Knowledge Base*

Ontology population is the process of adding content to a knowledge base by instantiating the classes defined in the underlying ontology. This section presents the approach I took to populate the knowledge base underpinned by HuCit which supports the extraction of canonical references. My approach aims to automate as much as possible this process by leveraging already existing resources such as CWKB and the Perseus Digital Library and by feeding back into the knowledge base, after manual verification, the results of mining canonical citations.

Importing Authority Data from the CWKB

The first resource I used to populate the knowledge base is the CWKB, which provides a large database of authority data about classical authors and works. Since CWKB now provides a LOD machine interface⁴⁶,

⁴¹ The knowledge base is accessible at <http://purl.org/net/hucit-kb>.

⁴² AllegroGraph, <http://franz.com/agraph/allegrograph/>.

⁴³ Lists of triple stores, maintained by the W3C, are available at <http://www.w3.org/wiki/SemanticWebTools> and <http://www.w3.org/wiki/LargeTripleStores>.

⁴⁴ SuRF – Object RDF mapper, <http://pypi.python.org/pypi/SuRF/>.

⁴⁵ Gruff: A Grapher-Based Triple-Store Browser for AllegroGraph, <http://franz.com/agraph/gruff/>.

⁴⁶ CWKB's LOD interface is available at <http://cwkb.org/linkedopendata>.

it was possible to harvest programmatically its content consisting of approximately 3,400 name variants for over 1,500 unique authors and over 6,500 work variants for 5,200 unique works. This resource made it possible to add to the knowledge base a substantial amount of authority data in the format described in section 3.5.5. This data consists of instances of authors (F1_Person), together with their name variants (E41_Appellation) and CTS URNs (E42_Identifier), as well as instances of their works (F2_Work) with their titles (E35_Title) and CTS URNs (E42_Identifier).

```

1  @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
2  @prefix hucit: <http://purl.org/net/hucit#> .
3  @prefix ecrm: <http://erlangen-crm.org/current/> .
4  @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
5  @prefix efrbroo: <http://erlangen-crm.org/efrbroo/> .
6  @prefix owl: <http://www.w3.org/2002/07/owl#> .
7
8  <http://purl.org/net/hucit-kb/authors/urn:cts:latinLit:phi0690>
9    a efrbroo:F10_Person ;
10   ecrm:P1_is_identified_by
11     ↪ <http://purl.org/net/hucit-kb/authors/678#cts_urn>,
12     ↪ <http://purl.org/net/hucit-kb/authors/678#name> ;
13   owl:sameAs
14     ↪ <http://data.perseus.org/catalog/urn:cts:latinLit:phi0690/>,
15     ↪ <http://cwkb.org/author/id/678/> .
16
17 <http://purl.org/net/hucit-kb/authors/urn:cts:latinLit:phi0690#name>
18   a efrbroo:F12_Name ;
19   ecrm:P139_has_alternative_form
20     ↪ <http://purl.org/net/hucit-kb/authors/678#abbr> ;
21   rdfs:label "P. Vergilius Maro"@la, "P. Virgilius Maro"@la,
22     ↪ "Publio Virgilio Marone"@it, "Publio Virgilio Marón"@es,
23     ↪ "Publius Vergilius Maro"@la, "Publius Virgilius Maro"@la,
24     ↪ "Vergil", "Virgil"@en, "Virgile"@fr .
25
26 <http://purl.org/net/hucit-kb/authors/urn:cts:latinLit:phi0690#abbr>
27   ecrm:P2_has_type
28     ↪ <http://purl.org/net/hucit-kb/types#abbreviation> ;
29   a ecrm:E41_Appellation ;
30   rdfs:label "Verg." .
31
32 <http://purl.org/net/hucit-kb/authors/urn:cts:latinLit:phi0690#cts_urn>
33   ecrm:P2_has_type <http://purl.org/net/hucit-kb/types#CTS_URN> ;
34   a ecrm:E42_Identifier ;
35   rdfs:label "urn:cts:latinLit:phi0690" .

```

Figure 3.27: Knowledge base example: the record for Vergil expressed as Turtle RDF.

Figure 3.27 provides an example of the instances contained in the knowledge base, specifically the RDF description for Vergil. It is worth highlighting the use of the property `owl:sameAs` (figure 3.27, line 11) to state in a machine-understandable way that the Perseus Catalog and CWKB provide as well descriptions of the same resource (i.e. Vergil).

The CTS URNs for authors and works, which are derived from the CWKB data, enable us to gather further information by querying Perseus' CTS API and to create further instances of HuCit classes as I describe next.

Importing Data from Perseus' CTS API

The second resource I used to populate the knowledge base is the Perseus Digital Library. Since the canonical divisions of texts are encoded as markup elements within the digital editions and translations contained in Perseus, it is possible to leverage such information in order to instantiate the relevant HuCit classes (i.e. `TextStructure` and `TextElement`). This operation can be fully automated given that the content in Perseus is accessible programmatically by using its CTS API, as already described in section 3.4.3.

The process involves gathering two pieces of information for each work contained in the knowledge base *and* in Perseus: first, information about the hierarchical structure of the canonical text divisions (e.g. the book/line structure of the *Aeneid*); second, a list of all the citable elements that make up such a structure (e.g. a list of all the books and lines in the *Aeneid*).

The first piece of information can be derived from the CTS method `GetCapabilities`, which returns a catalog with various metadata about the texts contained in the Perseus repository (figure 3.28). This catalog retains information about the canonical citation scheme of each available edition or translation of a work (lines 5–9). Instances of `E55_Type` are created based on such information and will be then associated to the corresponding `TextElement`. In the case of the *Aeneid*, the script adds the types “book” and “line” in case they are not already present in the knowledge base.

The second piece of information – a list of all books and lines in the *Aeneid* – is obtained by calling the `GetValidReffs` method with the CTS URN of the *Aeneid* as input parameter. This method returns an ordered list of lines that can be turned into `TextElement` instances, which in

```

1  <edition urn="urn:cts:latinLit:phi0690.phi003.perseus-lat1">
2    <label xml:lang="eng">Aeneid</label>
3    <description xml:lang="eng">Perseus:bib:oclc,22858571,
      ↳ Vergil. Bucolics, Aeneid, and Georgics Of Vergil. J. B.
      ↳ Greenough. Boston. Ginn & Co. 1900.</description>
4    <online docname="1999.02.0055.xml">
5      <citationMapping>
6        <citation label="book" xpath="" scope="">
7          <citation label="line" xpath=""
8            ↳ scope=""/>
9        </citation>
10     </citationMapping>
11  </online>
12 </edition>

```

Figure 3.28: The XML reply of the Perseus' CTS API upon a GetCapabilities request (some details omitted for the sake of readability).

turn are associated with the corresponding element type (figure 3.29, lines 5–9). The order of lines is preserved and recorded by means of the HuCit properties follows and precedes. Moreover, information about the book to which a given line of the *Aeneid* belongs is extracted from its URN and stored in the contains property.

Following this approach I was able to populate the knowledge base with the canonical text structures of 684 works (out of 5,200). To put this figure into context it is worth recalling that not all works contained in the knowledge base have a corresponding edition or translation in Perseus. Moreover, since the Perseus' CTS API is still in a development stage, the interface fails to return the requested information in 495 cases out of 1,179.

A final remark concerns the large volume of RDF triples that are generated through this process as this may challenge the scalability of the underlying triple store. For example, the *Aeneid* alone, with its 12 books for a total of approximately 9,700 lines, led to generating some 100,000 triples. The large number of triples, which is partly due also to the verbosity of the RDF language, is mostly caused by the fine level of granularity at which the canonical structures of texts are described.

Feeding the Results of the Citation Extraction back into the Knowledge Base

The third and last approach I took to the issue of ontology population consists of feeding the results of the automatic extraction of canonical references back into the knowledge base. However, given that the facts

```

1  @prefix ecrm: <http://erlangen-crm.org/current/> .
2  @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
3  @prefix hucit: <http://purl.org/net/hucit#> .
4
5  <http://purl.org/net/hucit-kb/types/book> a ecrm:E55_Type;
6      rdfs:label "book" .
7
8  <http://purl.org/net/hucit-kb/types/line> a ecrm:E55_Type;
9      rdfs:label "line" .
10
11 <http://purl.org/net/hucit-kb/works/urn:cts:latinLit:phi0690
12   ↪ .phi003:1> a hucit:TextElement;
13     rdfs:label "P. Vergilius Maro, 'Aeneid': book 1"@en;
14     ecrm:P1_is_identified_by
15       ↪ <http://purl.org/net/hucit-kb/works/urn:cts:latinLit:phi0690
16         ↪ .phi003:1#cts_urn>;
17     ecrm:P2_has_type <http://purl.org/net/hucit-kb/types/book> .
18
19 <http://purl.org/net/hucit-kb/works/urn:cts:latinLit:phi0690
20   ↪ .phi003:1.1> a hucit:TextElement;
21     rdfs:label "P. Vergilius Maro, 'Aeneid': book 1, line 1"@en;
22     ecrm:P1_is_identified_by
23       ↪ <http://purl.org/net/hucit-kb/works/urn:cts:latinLit:phi0690
24         ↪ .phi003:1.1#cts_urn>;
25     ecrm:P2_has_type <http://purl.org/net/hucit-kb/types/line> .
26
27 <http://purl.org/net/hucit-kb/works/urn:cts:latinLit:phi0690
28   ↪ .phi003:1.2> a hucit:TextElement;
29     rdfs:label "P. Vergilius Maro, 'Aeneid': book 1, line 2"@en;
30     ecrm:P1_is_identified_by
31       ↪ <http://purl.org/net/hucit-kb/works/urn:cts:latinLit:phi0690
32         ↪ .phi003:1.2#cts_urn>;
33     ecrm:P2_has_type <http://purl.org/net/hucit-kb/types/line> .

```

Figure 3.29: Knowledge base: the instances corresponding to lines 1-2 of *Aeneid*, book 1 expressed as Turtle RDF.

and statements contained in the knowledge base are assumed to be true, this approach always requires that the results of the automatic extraction are checked manually before they can be added to the knowledge base.

Although the manual correction of the results is not feasible in many situations due to limits of time and resources, it is an inevitable task when creating an annotated dataset as I did with a sample of abstracts drawn from the APh (see section 4.3.2). After correcting manually the results of the automatic processing, performed on some 360 abstracts for a total of approximately 25,000 words, I checked (programmatically) for cases where the names, titles and abbreviations present in the an-

notated data were not already contained in the knowledge base. This way it was possible to add 112 name variants, 134 title variants and 52 abbreviations to the knowledge base.

Similarly, this approach can also be applied to add `TextElement` and `TextStructure` instances. Consider for example the automatically extracted and manually verified citation *HG 7, 1, 44–46*, which refers to Xenophon's *Hellenica*. Let us assume also that the canonical text structure of this work is not yet present in the knowledge base. There is some useful information that can be easily deduced from the citation scope (i.e. 7, 1, 44–46). The citation reflects a hierarchy of three levels and implies that element "7" contains element "1", which in turn contains elements "44–46". This information allows us to create the corresponding `TextElement` instances and to fill the properties that represent the hierarchical relations between. What instead cannot be derived from this citation, without the input from a domain expert, is the type to be assigned to each element or, in other words, the fact that "7" refers to a book, "1" to a chapter and "44–46" refers to sections.

3.6.3 *Uses of the Knowledge Base*

The goal of the knowledge base is to provide a computer programme with the information about classical texts required to interpret correctly the canonical references automatically extracted from text. Let us now see in which ways – both current and future – such a knowledge base can be harnessed to improve the accuracy with which these references are extracted and interpreted.

Generation of Dictionaries

The knowledge base is used in the first place to generate dictionaries – i.e. lists – of names, titles and respective abbreviations that are used at various stages of the extraction of canonical references. The dictionaries of abbreviations are employed in the process of splitting texts up into sentences and then into tokens. Their use allows us to prevent some errors that are commonly caused by the presence of punctuation within abbreviations. Such dictionaries are of particular importance for the extraction of information – i.e. author names, work titles and canonical references – from texts written in several European languages as they

enable the citation extraction system to relate different spelling variants to the same entity.⁴⁷

Disambiguation of Implicit References

Another purpose of the knowledge base is to provide information that is not possible to deduce from a citation – or the context where it appears – and is needed to determine the precise meaning of the citation. A notable example of this situation is the omission of the title in a citation when the work being cited is the only one produced by the author or constitutes an opus maximum. In such cases, the indication of the author is sufficient to the trained reader to determine which specific work is meant in that context. The extraction system, instead, needs to query the knowledge base to get a list of the works by that author and, in case there is more than one result, selects the work marked as the author's opus maximum, provided that this information is contained in the knowledge base.⁴⁸

Another case where a knowledge base becomes useful, which was discussed at the beginning of this chapter in section 3.1, is to expand those citations that span several sections of a text (e.g. Plato *Rep.* 595a–596a). Such implicit references can be resolved if the knowledge base contains instances of all the citable text elements of a given work.

Validation of Extracted References

Since the knowledge base contains a list of all passages of a given text that can be cited, it could be used in the future to check the validity of the automatically extracted references. Such a validation is useful to detect those citations that, despite looking like plausible citations, are impossible given the actual structure of the cited text.

Consider the example reference “Hom. *Il.* 1.10.1”: such reference is impossible because the canonical text structure to cite the *Iliad* consists of two – not three – hierarchical levels (i.e. book/line). The (trained) reader would immediately spot the invalid citation. Instead, the citation extraction system needs to acquire some information about the canonical text structure of the *Iliad* from the knowledge base to determine that the string “.1” after “1.10” can not possibly be part of the citation. Similar mistakes are committed relatively often by the citation extrac-

⁴⁷ I discuss this topic in more detail in section 4.4.

⁴⁸ See section 3.5.5 on how such information is encoded in the knowledge base.

tion system when working with plain text OCR and may be due to the fact that footnote numbers are not tagged and therefore tend to be interpreted as part of the main text.

Disambiguation of Ambiguous References

Future research could focus on how the information contained in the knowledge base concerning the canonical text structures of classical texts helps improving the accuracy of the disambiguation of the extracted references. In fact, the number and type of hierarchical levels forming the canonical structure of a particular text constrain the set of possible valid references to that text.

Consider as example citation Th. 1.33.⁴⁹ The problem of disambiguating this citation is to determine whether it refers to book 1, chapter 33 of Thucydides' *Histories* or to line 33 of the first Idyll of Theocritus. In the first place, the extraction system looks up the citation string "Th." in the knowledge base to get a list of possible candidates. The results of this query can be further refined by looking at the canonical text structures of the candidates. The citation Th. 1.33 implies a canonical text structure consisting of two levels, and specifically implies the existence of has a first-level element with value "1" and a second-level element with value "33". A further query that looks for texts with these characteristics will help to narrow down further, but not enough, the set of candidates. These are book 1, chapter 33 of Thucydides' *Histories* – although "Th." is by no means the common abbreviation for the Greek historian – and Theocritus' Idyll 1, line 33. At this point one may turn to the context of the citation searching for clues. If the context contains the word 'idyll' or 'line/verse' the citation is most probably referring to Theocritus' work, whereas if it contains words such as 'book', 'chapter' or 'section' it is very likely to be a citation of Thucydides' *Histories*. Such a list of clues can be extracted from the knowledge base, and specifically from the labels of the different types of TextElement instances (see section 3.5.4).

⁴⁹ Crane et al. (2009, par. 26) provide this fitting example of the challenges that are faced when extracting and disambiguating canonical references.

3.7 SUMMARY

Canonical references are expressed using a human-readable notation. Making these references computationally tractable requires us to define a formal, computational model of their meaning. The HuCit ontology, which was presented in this chapter, serves this specific function and formalises the model that our practices of citing classical texts already imply (Smith, 2009). Moreover, the ontology serves as the model for a knowledge base – i.e. a database – containing the facts that a computer programme needs in order to correctly interpret the extracted canonical references. The main benefit of formalising this model with HuCit is that it enables us to publish in a semantic – i.e. machine-understandable – way the result of mining canonical citations as well as the information contained in the knowledge base. Once published in this way, information can be easily aggregated and integrated into new contexts, such as virtual reading environments or digital commentaries.

AUTOMATIC EXTRACTION OF CANONICAL CITATIONS

Overview

This chapter describes the approach I have developed to the automatic extraction of canonical citations. This approach consists of applying and adapting methods and techniques developed in Computer Science (CS) to the domain of Classics, and specifically to the problem of capturing references to ancient authors and texts. For this reason, in section 4.1 I introduce the key concepts of information extraction and the measures used to evaluate the results of this extraction. In section 4.2 I explain how my approach relates to past and ongoing research in the two sub-fields of CS, Information Extraction (IE) and Natural Language Processing (NLP). Section 4.3 describes the datasets that I created and mined for citations as part of this study. In section 4.4 I provide a detailed description of the sequence of steps involved in processing the data, the *pipeline*. Finally, in section 4.5 I discuss the results of the evaluation and consider ways in which the accuracy of the results could be improved.

4.1 KEY CONCEPTS IN INFORMATION EXTRACTION

4.1.1 *Extraction of Named Entities*

My approach to the extraction of canonical citations consists of adapting to the domain of Classics methods and techniques developed in CS to extract information from text. Therefore, before discussing in detail the extraction of citations, I explain how IE systems function by examining the working principles of a question answering system – i.e. a NLP

application that tries to answer questions formulated by users in natural language.¹

Consider the following passage drawn from the Wikipedia article on Aaron Swartz:

Two days after the prosecution rejected a counter-offer by **Swartz**, **he** was found dead in his **Brooklyn, New York** apartment, where he had hanged himself.

What are the steps required for a fictitious system to answer the question “Where did Aaron Swartz die?”.² First, the named entities (highlighted in bold) need to be captured – i.e. “Swartz” and “Brooklyn, New York”, respectively the name of a person and of a geographical place. This operation, called Named Entity Recognition (NER), aims to gather the facts that enable the system to provide an answer to the question above. Second, one needs to make explicit what relations exist between parts of the text – co-reference resolution and relation detection – and what entities are referred to by names – Named Entity Disambiguation (NED).³

In this case, disambiguating the named entities means establishing that the name “Swartz” refers to Aaron Swartz the programmer and activist, not the actor, while resolving the co-references requires linking the name “Swartz” to the pronouns “he” and “his”. The assertion “was found dead in [...]” constitutes a relation existing between “Swartz” (the person) and “Brooklyn, New York” (the name of the place where he died), which also needs to be captured.

As I will articulate in section 4.4, I have applied this three-step process to the extraction of canonical citations. To capture such citations and their meaning I first extract from text the citation components, then I determine the relations that exist between these components and finally I disambiguate the extracted citations by assigning each of them the corresponding unique identifier.

¹ For a more comprehensive and detailed explanation of the concepts discussed in this section I refer the reader to Jurafsky and Martin (2009).

² I have deliberately left out the preprocessing steps, such as sentence segmentation and tokenisation, since they are of minor importance and will be addressed in section 4.4.

³ It can also be referred to as *named entity resolution*, *entity linking* or *normalisation*.

4.1.2 *Methods in Natural Language Processing*

The problem of extracting named entities from text, as any other NLP task, can be approached using different paradigms. The rule-based approach involves defining and applying a set of rules that signal the presence of a name (e.g. a name is a sequence of alphabetic characters that starts with a capital letter). In contrast, the machine learning approach consists of training a model to learn from a set of input examples what features characterise a named entity. The system I have developed to extract canonical references uses a mixture of rule-based and machine learning algorithms.

When talking about machine learning methods it is important to distinguish between *supervised* and *unsupervised* methods. The difference between these two methods is that supervised learning requires annotated and manually corrected data. There are also weakly supervised and semi-supervised methods, which both aim to reduce the amount of annotated data that is needed to train the model.

In this study I have adopted a supervised machine learning approach and applied it to the identification of named entities, a task consisting of assigning the correct named entity type to each word in a sentence. Supervised methods have been extensively applied to NLP tasks that entail some kind of classification such as detecting spam emails or tagging named entities. Supervised methods are suitable in cases where the expected output is already known as is the case with named entity recognition, which consists of assigning to each word in a sequence the corresponding named entity type drawn from a predefined set. Unsupervised methods are employed when one wants to elicit some patterns from the input data without making any assumptions concerning the output.

In my case I chose a supervised approach because when extracting named entities the expected output is already known. The main advantage of using such an approach is that the model can be re-trained with different training data. In the case of extracting citations, this means that the system can be trained to cope with the citation style characterising a specific set of documents. For example, given that different journals may differ as to how classical texts are cited, the system could be trained to capture the specificities of the citation style used by a given journal.

Supervised learning consists of training a statistical model to perform a given classification task such as assigning to each word in a sentence the corresponding named entity type. In order to predict the output, the model needs to be fed with training data where each instance to classify is associated with the corresponding label. By observing a number of features extracted from the training data the model is able to assign to each possible label a score indicating the probability of that label being assigned.

Three kinds of data are typically used in a supervised learning setting: the *training set*, which consists of data that is used to compute the probabilities; the *testing set*, which consists of unseen data used to evaluate the overall performance of the system (i.e. data that is not part of the training set); the *development set*, which consists of data different from the training and testing set. The development set is used to avoid the problem of overfitting, a situation in which the trained model performs poorly on data different from the training set. In section 4.3 I discuss the creation of an annotated training and test set for the specific task of extracting canonical citations from text.

4.1.3 Evaluation Metrics: Precision, Recall and F_1 Score

This section introduces the metrics that have been developed to evaluate the accuracy of information extraction systems. These metrics are used in section 4.5 to evaluate the system I have implemented to extract canonical citations.

NLP research is characterised by a strong focus on the evaluation of the various algorithms that can be employed to perform the same task. Since 1999 the Conference on Natural Language Learning (CoNLL) has been organising *shared tasks*, namely co-located events usually centered around one specific task (e.g. NER) in which the different systems developed by the participants are evaluated against the same data that is provided by the organisers.⁴ Some of these events, however, focus on specific disciplines: the 2013 Association for Computational Linguistics (ACL) conference, for example, featured the shared task “Corpus Annotation with Gene Regulation Ontology” which dealt with one specific aspect of extracting information from biomedical literature.

⁴ CRF++, <http://takug910.github.io/crfpp/>.

Error types

A key concept of evaluation is the classification of errors into different types. The error types vary in relation to the specific task that is being evaluated. In the following example I shall illustrate the types of errors that are encountered when evaluating the extraction of named entities. Let us suppose that our NER system has produced the output shown in figure 4.1. A comparison of the expected with the actual output shows that: the entity “Swartz” was correctly classified and is therefore a true positive (TP); “Two” was incorrectly recognised as a named entity thus constituting a false positive (FP); the entity “Brooklyn, New York” was missed by the system and is therefore a false negative (FN); all the other words that were correctly classified as not being named entities count as true negatives (TNs).

Evaluation of named entity extraction

Expected output:

Two days after the prosecution rejected a counter-offer by Swartz, he was found dead in his Brooklyn, New York apartment, where he had hanged himself.

Output:

Two days after the prosecution rejected a counter-offer by Swartz, he was found dead in his Brooklyn, New York apartment, where he had hanged himself.

Figure 4.1: Comparison of expected and obtained output for an example sentence showing the four error types: true positives are highlighted in green, false positives in yellow and false negatives in red; unhighlighted words are true negatives.

How do we transform these observations into numeric values that allow for comparison across different systems? This is done by means of standard metrics – *precision*, *recall* and *F₁ score*. These metrics take into account different combinations of the error types illustrated above in order to quantify slightly different aspects of the accuracy of the evaluated system. Figure 4.2 illustrates the four error types and how they come into play when calculating the precision and recall metrics, which are introduced next.

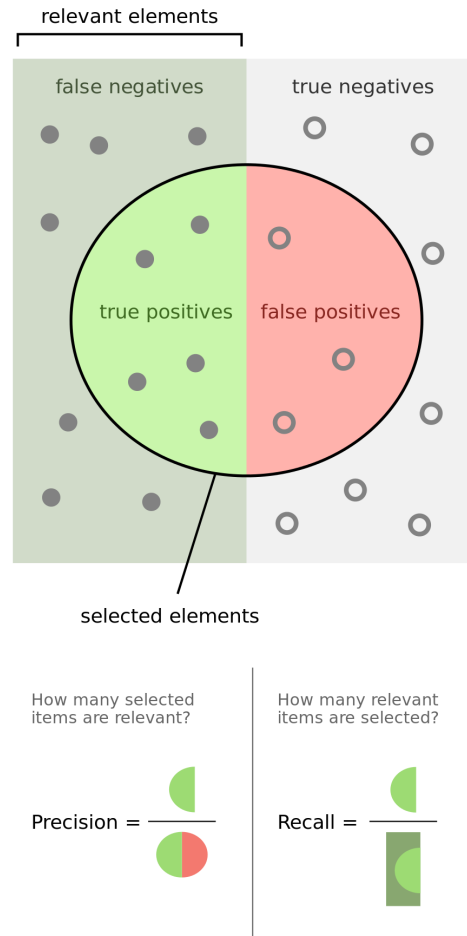


Figure 4.2: Schematic illustration of the differences between the four error types and their relation to precision and recall from Walber (Own work) [CC-BY-SA-4.0], via Wikimedia Commons.

Precision

$$\text{Prec} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4.1)$$

Precision is defined by the formula in equation (4.1) and is the fraction of retrieved entities that are correct. This measure takes into account the correctly identified entities (TPs) as well as those that were mistakenly tagged as entities (FPs), but does not consider the entities that were missed (FNs). The precision of a system that produces the output of figure 4.1 is calculated as $\text{prec} = \frac{1}{1+3}$ and is therefore 0.25 (or 25%).

Recall

$$\text{Rec} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4.2)$$

Recall is defined by the formula in equation (4.2) and is the fraction of correct entities that are retrieved by the system. Recall does not consider the FPs but, instead, takes into account the TNs, i.e. the entities that were missed – in this case the place name “Brooklyn, New York”. The recall given the output above is calculated as $\text{rec} = \frac{1}{1+1}$ and is 0.50 (or 50%) – in fact, out of two entities (i.e. “Swarz” and “Brooklyn, New York”) only one was correctly extracted.

F₁ Score

$$F1 = \frac{2PR}{P + R} \quad (4.3)$$

Finally, the F_1 score (or F-measure) defined by the formula in equation (4.3) is a global metric that combines both precision and recall giving them equal importance. It is the weighted harmonic mean of precision and recall where the weight is expressed by the coefficient β : with $\beta < 1$ the precision is favoured, while with $\beta > 1$ the recall is favoured; a value of $\beta = 1$ treats them equally. F_1 indicates that the weight being used in the equation is 1. Therefore, the F_1 score of the system that produces the output above, given a precision of 0.25 and a recall of 0.50, would be $F1 = (2 * 0.25 * 0.50) / 0.25 + 0.50$ thus 0.33 (or 33%).

4.2 THE LANDSCAPE OF INFORMATION EXTRACTION RESEARCH

This section aims to situate my research on the automatic extraction of canonical citations within the broader landscape of research on IE. First, I look at research on this topic in disciplines other than Classics. I then focus on domain-specific named entities and discuss some applications from Archaeology as well as from some more distant fields such as Biomedicine and Legal Studies. I conclude with a section on the extraction of modern bibliographic references by considering the similarities in terms of issues raised and solutions applied between this problem and the extraction of references to primary sources.

4.2.1 *Information Extraction Systems*

The origins of research on IE are found outside academia. IE systems were developed in the 1980s by the US Navy initially with the goal of

extracting information from news and military texts (Grisham, 2010). However, in the last decade due to large scale digitisation initiatives and to the growing volume of publications being made available online the problem of finding scalable solutions to extracting information has become increasingly important in a number of disciplines. I now consider some IE systems that were developed both within and outside of the disciplines of Classics and Archaeology that are relevant to my approach.

Biomedicine, and specifically Biomedical Natural Language Processing (BioNLP), has been heavily investing in automatic information extraction. The proceedings of recent BioNLP workshops, which were co-located with the ACL conference, give an idea of the depth and breadth of research in this area (Cohen et al., 2013,0). The range of tasks covered spans the extraction of gene and protein names from biomedical texts to tasks focussed on very specific problems such as the shared task *Cancer Genetics*, which deals with the extraction of information from literature on cancer.⁵ Despite the distance between Classics and Biomedicine, my research benefitted from the results produced in this area in two respects: first, the Brat Rapid Annotation Tool (Brat) that I employed to annotate the data was developed within the BioNLP community (Stenertorp et al., 2012); second, the tool that achieved the greatest accuracy with regards to sentence segmentation – the task of splitting a text into sentences – was the one that is included in Brat.⁶

Further examples of information extraction systems can also be found in disciplinary areas closer to the Humanities. In the Fine Art domain, Odat et al. (2015) present a system for the extraction of information related to chemical processes and preservation treatments from unstructured texts related to the conservation of paintings. The solution they propose closely resembles the approach I am adopting. First, the system is based on an ontological *knowledge base* containing key concepts that describe paintings as well as the methods and techniques for their conservation. Second, a combination of machine learning-based and rule-based algorithms are used to capture the named entities and the relations existing between them. Third, entities and relations, once extracted from text, are mapped onto concepts and properties contained

⁵ Cancer Genetics (CG) task, <http://2013.bionlp-st.org/tasks/cancer-genetics>.

⁶ See section 4.4.1 for the problematic aspects of sentence segmentation when dealing with texts from Classics.

in the knowledge base. Finally, the results of the extraction allow users to search for publications related to a specific technique or issue in the art conservation domain.

Similar systems were also devised in the field of Archaeology with the goal of distilling specific pieces of information from unstructured texts. Paijmans and Wubben (2007), for example, focussed on the automatic indexing of archaeological papers and reports. They used a machine learning-based approach to capture chronological and geographical references and measurements. Following that, Byrne and Klein (2010) developed a system aimed at extracting event structures such as archaeological finds, excavations and surveys from text in order to transform the extracted information into semantic data.

Recent research in the field of Digital Classics has focussed on the extraction of one specific named entity type, namely geographic place names. This task is commonly known as *geoparsing* and, similar to what happens with other kinds of named entities, consists of the two separate steps of extracting and resolving place names.⁷ *Geoparsing*, also known as *geotagging*, deals with identifying chunks of texts that constitute place names, whereas *georesolution* consists of determining the most likely geographical location of each extracted place name.

The *geoparsing* of digitised texts was the main goal of the recently concluded Google Ancient Places (GAP) project (Isaksen et al., 2012). The project relied on the Edinburgh Geoparser for the *geotagging* step (Grover et al., 2010), while the *georesolution* was carried out by linking place names to matching entries in the Pleiades gazetteer of ancient places.⁸ The efforts initiated by GAP are now being continued by the project Pelagios⁹, which stands for Pelagios Enable Linked Ancient Geo-data In Open Systems. Pelagios, which is in its third funding phase at the time of writing, also deals with *geoparsing* but with a specific focus on primary sources and tries to cover a wide range of traditions, both geographically and chronologically (Simon et al., 2012).

⁷ For a broader reflection on the history of the relationship between geography and computing in Classics see Elliott and Gillies (2009).

⁸ Pleiades, <http://pleiades.stoa.org/>.

⁹ Pelagios, <http://pelagios-project.blogspot.de>.

4.2.2 *Citations and other Discipline-specific Named Entities*

As pointed out in section 1.5, the idea of modelling canonical citations as named entities was first suggested by Crane et al. (2009). This approach follows the more general trend in NLP and NER research of extending the hierarchy and number of named entities to cover discipline-specific pieces of information (Nadeau and Sekine, 2007, pp. 2–4). This trend leads to the identification of more finely grained entities that are of interest to scholars within a specific domain: some examples of such domain-specific entities are discussed next.

The named entities that are common to both the Message Understanding Conference (MUC) and CoNLL sets are Location, Person and Organization.¹⁰ The MUC set also contains the category Misc, which captures entities not falling into any of the categories above. CoNLL adds to the core set 4 more entities: Time, Money, Percent and Date. In addition to these general entities there exist named entities that capture concepts of interest within specific domains. The mentions of genes and proteins, for example, are considered as named entities in the field of BioNLP (Settles, 2004), while entities such as Judge, Attorney, Company, Jurisdiction and Court, which are more finely grained instantiations of the entities Person and Organisation, are useful when extracting information from legal texts.

The idea of treating references as if they were named entities is not totally unprecedented. Francesconi et al. (2010) extracted case citations from legal texts while Galibert et al. (2010) extracted references to patents. In both cases the identification of such references is necessary to create networks of citations between documents that allow for an effective means of finding information within large sets of legal or patent-related documents. The approach I have adopted, however, differs from these approaches insofar as I model a citation as a relation between citation components instead of treating it as a single, monolithic entity. In section 4.3.1, I explain my rationale for making this decision and discuss, by means of examples drawn from the annotated data, the benefits of my approach.

¹⁰ A fixed-space font is used throughout this chapter to indicate named entity types.

4.2.3 *The Extraction of Modern Bibliographic References*

Although the main focus of this dissertation is on canonical references there are other kinds of bibliographic references that can be extracted from texts such as those to modern publications. In this section I shall discuss research on the extraction of references to modern publications that is especially relevant in the context of this research, with a focus on the issues that are raised when working with humanities publications.

The extraction of references to secondary sources such as monographs, edited volumes and journal articles is an essential task to perform when creating citation indices like those that were considered in section 1.4.2 above. This task consists of two distinct operations: firstly, locating the references within the document and, secondly, parsing each reference in order to identify the various pieces of bibliographic information (e.g. author, title, publication date, etc.). In particular, making this process as automatic as possible is of great importance for the scalability of such endeavours, i.e. their ability to handle large sets of documents. As a result, there has been extensive research on this topic in CS over the last two decades starting with the seminal work by Giles et al. (1998) on an automatic citation indexing system for Citeseer. Citeseer is a citation index that primarily covers literature in CS and Library and Information Science (LIS). The current version of this index, called CiteSeer^x, relies on ParsCit¹¹ for the extraction of bibliographic references. My approach to extracting canonical citations, and particularly the feature set used for training the statistical model, was informed by the work done by Councill et al. (2008) on the development of ParsCit (see section 4.4.2).

Furthermore, the high level of accuracy that current citation extraction systems are able to achieve has certainly contributed to their application in a variety of scenarios. The platform ResearchGate¹², for example, uses an indexing system called Grobid¹³ in order to extract automatically bibliographic references from the full text of publications that are authored by the users and uploaded to the platform (Lopez, 2009). One feature of this platform, which is enabled by such an indexing system, is the ability to connect with researchers that cite one's own publications.

¹¹ ParsCit, <http://aye.comp.nus.edu.sg/parsCit/>.

¹² ResearchGate, <http://researchgate.net/>.

¹³ Grobid, <http://github.com/kermitt2/grobid>.

Although several tools and services are available for the extraction of bibliographic references, the accuracy of these tools tends to worsen when they are applied to humanities publications. This is due to the fact that, up until now, most of the development of automatic citation indexing systems has taken place in the scientific domain. As a result, certain discipline-specific citation practices result in difficulties for these tools. A notable example is the extraction of those expressions which are usually found in footnotes and are used to refer to publications that have already been cited (e.g. “idem”, “ibidem”, “op. cit.”, “loc. cit.”, etc.). This issue is ultimately one of anaphora resolution: what needs to be established is to which one of the already cited items the anaphoric expression refers to (e.g. “idem”). In order for the anaphora to be resolved one needs to isolate similar expressions and maintain the precise order in which publications are cited in the text.

4.3 CREATION OF ANNOTATED DATASETS

Defining an annotation scheme is an essential part of the broader process of translating a specific problem or phenomenon into computational terms. Modelling the extraction of canonical citations as a named entity recognition task requires entities and relations to be established. The entities and relations that I have specified in my research in order to annotate canonical citations within the data manually and automatically are defined by the annotation scheme described in section 4.3.1.

In section 4.3.2 I provide some detailed information about the datasets that were used in this research – L’Année Philologique and JSTOR. Finally, the file formats that are used to store the annotated datasets are briefly illustrated in section 4.3.3. Understanding the scheme that was used to annotate the data is essential in order to understand the extraction pipeline described in section 4.4.

4.3.1 *A Scheme to Annotate Canonical Citations*

Named Entities

The annotation scheme comprises the following entities:

- Author: captures mentions of the name of an ancient author, e.g. “Xenophon”;

- Awork: captures the title of an ancient work, e.g. “Hellenica”;
- Refauwork: captures a structured reference to the cited work, where the structure is typically created by the use of punctuation and abbreviations. Refauwork can be used to tag a reference to an author (e.g. “Thuc.” for Thucydides), to a work (e.g. “Hell.” for Xenophon’s *Hellenica*) or to tag a sequence in which both author and title are indicated (e.g. “Xen., Hell.”). This entity was also devised to cover those cases where the abbreviation of the author’s name is used, in a sort of metonymy, to refer to the *opus maximum* or the only work written by a given author (e.g. “Thuc.” for Thucydides’ *Histories*);¹⁴
- Refscope: is used to annotate the scope of the citation, i.e. the precise indication of which section of a given work is being cited (e.g. the portion “3.3.1–4” of the citation “Hell. 3.3.1–4”).

In designing this scheme I examined a wide variety of citation examples so as to ensure that it can be used to annotate virtually *any* canonical citation. These examples – some of which were already discussed in section 2.1 and section 3.2 – represent diverse citation styles and cover a variety of citation practices ranging from the early modern to the contemporary.

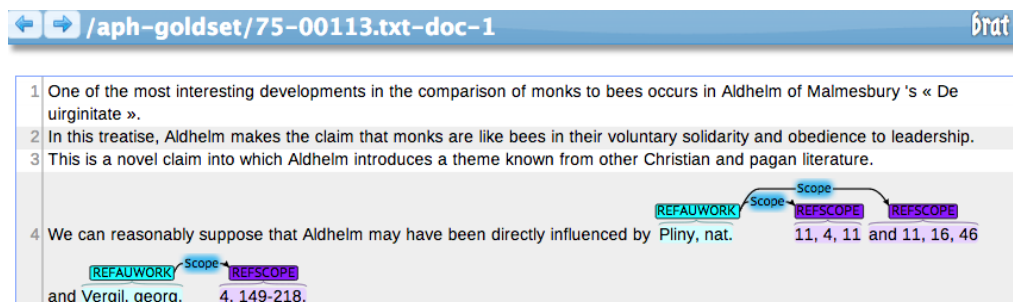


Figure 4.3: An example of annotated APh document (75-0113) visualised in Brat: the highlighted portions of text indicate named entities, while the arrows represent the relations existing between them.

The visualisations that are provided throughout this section, such as the one in figure 4.3, were produced using Brat, which is the environment of choice for annotating and visually displaying the data. In such visualisations the named entities are indicated as spans of texts of different colours with a label to indicate the entity type (e.g. Refauwork).

¹⁴ For a definition of *opus maximum* see *infra* p. 52.

The relations between entities are represented as arrows connecting two or more entities with a label indicating the type of relation (e.g. *scope*).

Relations

Given these four entities, a canonical citation is defined as a binary relation between any two entities, where one entity must be by definition the indication of the citation's scope (*Refscope*) while the other can belong to any of the remaining entity types (*Aauthor*, *Awork* and *Refauwork*). figure 4.3 shows how the citations "Pliny, nat. 11,4,11 and 11,16,46" and "Vergil, georg. 4,149–218" can be represented as relations between their components.

There are two main reasons why it seemed preferable to model citations as relations between citation components rather than as single entities – as I had initially suggested elsewhere (Romanello et al., 2009c). This approach allows for better representing discursive citations, namely citations that are constituted by non consecutive tokens. An example of this kind of citation is provided in figure 4.4. Such a citation could not be captured by means of one single entity, which requires all tokens that constitute the citation to be consecutive. Instead, it can be represented as a relation between the author's name (i.e. "Ammianus") and the cited passages. It is also worth noting that the name "Ammianus" is used here in a metonymic way to refer to the work *Res Gestae*, his opus maximum.

The second reason for preferring this solution is that entities of the same type tend to be homogeneous in terms of the features they display: *Refscope* entities, for example, are mostly made of numbers and punctuation signs, whereas *Aauthor* and *Awork* entities almost never contain numbers and in most cases begin with a capital letter. When training a statistical model to recognise a set of entities, the model is more likely to achieve better results if each entity type is characterised by a relatively homogeneous set of features.

Disambiguation

In addition to extracting named entities and the relations between them, it is also important to make explicit what exactly is being referred to, an operation known as disambiguation. For example, the entity *Aauthor* identified by the string "Ammianus" in figure 4.4 refers to the ancient author Ammianus Marcellinus.

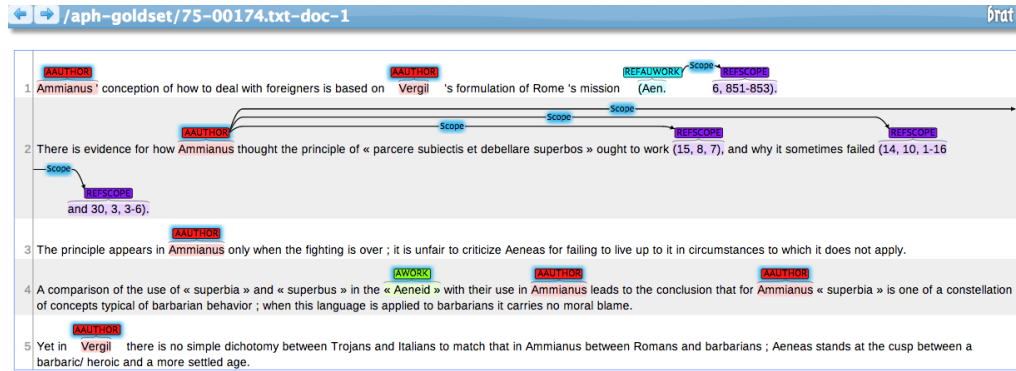


Figure 4.4: Representing canonical references as relations between named entities allows us to capture the discursive reference “Ammianus [...] (15, 8, 7) , and [...] (14, 10, 1-16 and 30, 3, 3-5)”, which is made up of non consecutive tokens.

A common approach to disambiguating named entities is to use links to Wikipedia pages as unique identifiers. This solution would be suitable when disambiguating ancient authors and their works but it would not be of much help when identifying specific sections of ancient works as there are no Wikipedia entries for each citable section of any ancient work. The solution I chose is to use the identifiers specified by the Canonical Text Services (CTS) protocol and based on the Uniform Resource Name (URN) syntax.¹⁵ For instance, Ammianus is identified by `urn:cts:latinLit:stoa0023`, his *Rerum Gestarum* has the CTS URN `urn:cts:latinLit:stoa0023.stoa001` and book 14, chapter 1 of this work can be referred to by the identifier `urn:cts:latinLit:stoa0023.-stoa001:14.1.1`.

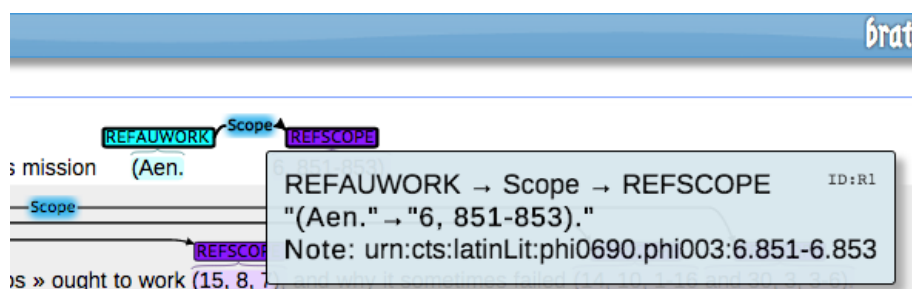


Figure 4.5: This figure shows how a CTS URN is attached to the scope relation it disambiguates and is visualised on mouseover in Brat.

These identifiers are stored within the comment field attached to any given entity or relation in the annotated datasets (see figure 4.5). They disambiguate named entities by pointing to individuals that are con-

¹⁵ For further information on the CTS protocol see section 3.4.3.

tained in the knowledge base discussed in the previous chapter. For example, in table 4.1 the name “Ammianus” is assigned the identifier `urn:cts:latinLit:stoa0023`. Once this identifier is looked up in the knowledge base it becomes possible to derive the information that the entity thereby identified has the ontological class `frbroo:F10_Person` or, in other words, is a person as defined by the FRBR Object-Oriented (FRBR_{OO}) ontology.

Table 4.1: Mapping between strings, CTS identifiers and ontology classes for the named entities and relations contained in APh 75–00174 (the prefix `urn:cts:latinLit:` was stripped off from all identifiers in order to make the table more easily readable).

String	Type	Identifier	Ontology Class
Ammianus	Aau- thor	stoa0023	frbroo:F10_- Person
Vergil	Aau- thor	phio690	frbroo:F10_- Person
(Aen. 6,851–853)	Scope	phio690.phio03:6.851–6.853	hucit:TextElement
Ammianus (15,8,7)	Scope	stoa0023.stoa001:15.8.7	hucit:TextElement
Ammianus (14,10,1–16)	Scope	stoa0023.stoa001:14.10.1– 14.10.16	hucit:TextElement
Ammianus (30,3,3–6)	Scope	stoa0023.stoa001:30.3.3–30.3.6	hucit:TextElement
Aeneid	Awork	phio690.phio03	frbroo:F1_Work

4.3.2 The Datasets: APh and JSTOR

This section aims to provide a rationale for choosing the datasets that I have used in my research and to highlight the differences between the datasets. Two corpora were identified for this purpose: one is made up of abstracts extracted from the analytic bibliography *L’Année Philologique* (APh) and the other consists of the journal articles in JSTOR that are related to Classics.

The selection was based on four criteria. First, the corpora had to be of essential importance to classical scholars. Given that the methods I have used in my research are likely to be unfamiliar to many classicists, I deliberately chose two well known resources. The second criterion was the size, which had to be challenging enough so as to justify an automatic approach to the data processing. Third, the chosen corpora needed to be significantly different in terms of the types of resources,

quality of the data and style of the citations they contained. Such differences were important in order to ensure the general applicability of the devised approach or, in other words, to avoid what in machine learning terms is known as *overfitting*, i.e. a situation in which a trained model performs well on the training set but performs poorly on unseen data. Fourth and last criterion was the language of the texts. The fact that English did not completely replace other European languages as a means of scholarly communication in the field of Classics makes covering as many of these languages as possible an important requirement.

In chapter 2 I discussed the importance of classical commentaries as a literary genre. However, I decided not to mine citations from them. The availability of commentaries is certainly not an issue as many of them, and particularly those that are already in the public domain, were recently made accessible in electronic format by large scale initiatives such as Google Books and the Internet Archive. However, the effort needed to prepare the data for processing made the mining of citations from digitised commentaries impracticable. In fact, the poor quality of the OCR of these texts, especially when they contain polytonic Greek, means that the OCR needs to be reperformed in order to bring the texts to a state that is actually suitable for information extraction purposes.

Another disadvantage of working with commentaries is the complex layout structure that such texts present, often consisting of multiple columns or horizontal levels as shown in the example commentaries discussed in section 2.1. Such a complex layout structure is likely to cause some problems with the OCR and would require some structural markup in order to distinguish the various levels of text in the printed page. Despite the technical challenges they present, commentaries are an extremely interesting type of texts to extract canonical citations from.

L'Année Philologique (APh)

The first corpus I worked with consists of texts drawn from the APh. The APh is a critical and analytical bibliography that has been published annually since 1924 and indexes virtually any publication that has some relevance to research in Classics. It may be said without exaggeration that it constitutes the starting point for any research in this field. For each indexed publication the APh provides a short abstract summarising the topic covered as well as the main points discussed.

One essential characteristic of the APh is the abundance of canonical citations. Not every APh abstract, however, contains such citations. The abstract of a publication focussing on Archaeology, for instance, is more likely to contain references to museum objects than citations of texts. On the contrary, summaries of publications that focus on text often contain an indication of the main text passages that the authors discussed, these are signalled in the text by means of canonical citations.

Since the work of the reviewers is informed by the same abstracting guidelines, the style of canonical citations tends to be fairly homogeneous throughout the APh. Work titles, for example, are always enclosed by angle quotes called Guillemets (“«” and “»”). Moreover, the abstracts are written in the main European languages – i.e. French, Italian, Spanish, German and English – depending on the national office where the indexing and abstracting was carried out. In fact, the work of the APh is organised by means of a distributed network of national offices that are responsible for abstracting the publications and inputting the data into a central system ensuring the thorough coverage of this resource.

The first sampling decision I had to take, given the scale of this resource, was to work on one of the 80 annually published volumes. The 2004 volume – volume 75 – was chosen and this resulted in a dataset of 354,672 tokens. Since this dataset was still too large for manual correction, I had to apply a second sampling step to reduce the number of tokens to a manageable size (25,889 tokens, see table 4.2).

Table 4.2: Number of documents and tokens of the APh dataset. The dataset is divided into two subsets: the development set consisting of documents that were annotated automatically and the training set consisting of documents that were also manually checked.

Subset	Documents	Tokens
development set	6,947	354,672
training set	366	25,889
Total	7,313	380,561

The dataset was first automatically annotated according to the annotation scheme discussed in the previous section. The resulting annotations were then manually verified by two domain experts. An annotated and manually corrected corpus was required in my research for two reasons: first, to be used as the baseline when evaluating the accuracy with

which canonical citations can be extracted automatically and, second, to be used as input data to train a statistical model to capture named entities from text. Since this dataset is the first of its kind and given that annotating the data is a highly time consuming process, considerable efforts were made to make the corpus openly available so as to facilitate further research on this topic.¹⁶

The sampling method I used to reduce the number of tokens that had to be manually corrected without sacrificing the effectiveness of the corpus is called Active Annotation. Active Annotation is the adaptation of the Active Learning method (Settles, 2009) to the task of creating an annotated corpus; it was first proposed by Vlachos (2006) with relation to the task of tagging named entities in biomedical texts and has recently found application to humanities data (Ekbal et al., 2011).

This method has been found to reduce the effort of creating training material, which is a crucial bottleneck when adapting supervised learning methods to a new domain. The idea underlying Active Annotation is that, in order to reduce the amount of training data required, the documents for manual correction are carefully selected instead of being randomly sampled. What exactly does *carefully selected* mean in this context? The instances (i.e. sentences) are chosen based on how informative they are for the classifier. The instances with which the classifier had the greatest difficulties in predicting the correct label are selected. The informativeness of a given instance is quantified by a parameter called the *confidence interval*, which measures the degree of confidence of the classifier in making a certain prediction.

Table 4.3: Basic statistics about the training set derived from the APh dataset with the documents grouped by language.

Subset	Documents	Sentences	Tokens
train-de	39	72	2,664
train-en	65	208	6,231
train-es	32	59	2,444
train-fr	123	230	6,633
train-it	107	130	4,374
Total	7,312	699	25,889

¹⁶ The APh dataset was released under an open source licence and is available at <http://dx.doi.org/10.5281/zenodo.12762>.

To sum up, Active Annotation was used to reduce the number of tokens to be manually annotated from 354,672 to 25,889. The breakdown of documents in the training set by language is shown in table 4.3, while table 4.4 summarises the number and type of annotations that were produced and manually corrected.

Table 4.4: Number and type of annotations contained in the APh training set.

Annotation	n
Aauthor	368
Awork	231
Refauwork	221
Refscope	441
Scope relation	381

JSTOR

A second dataset used while developing the citation extraction system consists of journal articles drawn from JSTOR. JSTOR is the most comprehensive single archive of journal articles related to Classics containing the full text of 138,000 articles belonging to some 1,380 journals for a total of more than 320 million tokens.¹⁷ I have used this dataset in order to guarantee that the system would also work on documents that differ from those contained in the training set or in other words to avoid overfitting.

The number of journals included and the timespan covered result in a wide diversity of citation styles. Processing a corpus of this scale represents a challenge in terms of time and computing power. At the same time, the scale makes JSTOR a unique resource to work with allowing us to track a given phenomenon, such as text reception, over a time span of more than two centuries.

The documents in my dataset are those articles contained in JSTOR that were classified as being related to Classics. Since the classification was performed automatically by using a clustering technique called Topic Modelling, the dataset does contain some documents that are not related to Classics and were incorrectly classified. As the content in JSTOR constantly grows and errors in the automatic classification may

¹⁷ Given the Optical Character Recognition (OCR) errors that are contained, this figure may not correspond to the number of actual tokens.

disappear, the dataset I have worked with needs to be understood as a snapshot of the data at a specific point in time.¹⁸

As far as the licence is concerned, for datasets of up to 1,000 documents the data can be freely obtained by using Data for Research¹⁹, an online tool that allows researchers to interact in various ways with JSTOR's content. If one needs to access a dataset that exceeds this limit, however, it is necessary to describe the research for which the data is being requested and, upon approval by JSTOR, to sign a licence agreement. This restricted licencing policy was, in fact, the main reason why the APh was preferred to JSTOR as a source of data to create a reusable dataset.

This corpus differs substantially from the one that was previously described in several respects. The first difference concerns the quality of the data: while the APh corpus consists of cleanly transcribed texts that were exported from a database, the data in the JSTOR corpus is often the result of OCR and because of this the quality is variable. Although the general approach was to accept the presence of OCR errors without trying to correct or recover them, in some cases dealing with these errors was unavoidable. This was the case, for example, with characters that were wrongly recognised by the OCR software and thus transcribed into random sequences of characters. Sometimes these characters would combine to form sequences that have a special function, thus causing problems with operations such as reading in the input files (e.g. the sequence “\n” indicates a new line).

Another difference is that the JSTOR articles come as plain text, i.e. without any kind of structural markup to distinguish between the different sections of an article (e.g. running headers, page numbers, footnotes, bibliography, etc.). What this means in practice is that, given that footnotes of articles in Classics tend to contain a wealth of canonical citations, without markup it is impossible to link such citations back to their original context. Moreover, running headers and page numbers become part of the text thus causing some noise that would be desirable to filter out. Similarly, given that citations to primary and secondary sources look relatively similar, not being able to isolate the bibliogra-

¹⁸ The dataset described in this section was created on November 26 2013 by JSTOR staff using the following request <<http://dfr.jstor.org/fsearch/submitrequest?fs=tom1:tgml&view=text&cc=subject:classicalstudies-discipline^1.0>>.

¹⁹ Data for Research, <http://dfr.jstor.org/>.

phy section from the rest of the article caused errors in the extraction of citations and named entities.

4.3.3 File Formats of the Datasets

Once annotated according to the scheme that was illustrated in section 4.3.1, the data was stored using two distinct file formats: the Input, Outside, Beginning (IOB) format and the standoff markup format used by the annotation environment Brat.

influenced	VBN	0
by	IN	0
Pliny	NP	B-REFAUWORK
,	,	I-REFAUWORK
nat.	NP	I-REFAUWORK
11	CD	B-REFSCOPE
,	,	I-REFSCOPE
4	CD	I-REFSCOPE
,	,	I-REFSCOPE
11	CD	I-REFSCOPE
and	CC	B-REFSCOPE
11	CD	I-REFSCOPE
,	,	I-REFSCOPE
16	CD	I-REFSCOPE
,	,	I-REFSCOPE
46	CD	I-REFSCOPE
and	CC	0

Figure 4.6: The annotation of a canonical reference represented in the IOB tabular format. The three columns contain respectively: the annotated token; its Part-of-speech (PoS) tag and the named entity label assigned to it.

IOB is a tabular format that is used in NLP for a variety of chunking tasks (e.g. NER). An IOB file, such as the one given in figure 4.6, contains one token per line with blank lines to indicate the boundaries between sentences; in turn, each line contains values that may be organised in multiple columns and are typically separated by means of the tab character (i.e. “\t”). The name of this file format derives specifically from the notation that is used to label the tokens: the letter 0 is used for tokens without a label; a label starting with prefix B- is used for the first token of an entity, whereas I- is used for all successive tokens.

The format used by Brat belongs to the category of standoff markup formats. The term *standoff markup* indicates that the annotations about the text – i.e. the named entity labels in the case of this study – are

stored separately from the text itself. An annotated span of characters is then identified by means of a numeric offset, i.e. a character index relative to the origin where the index of the first character of the document is 0. For example, the first row of the standoff markup displayed in figure 4.7 is to be read as “the span of text identified as T1 with start index 442 and end index 453, which corresponds to the string ‘Pliny, nat.’ has the label Refauwork”. A computer program that manipulates such annotations typically reads the text into an ordered list of characters – or array – then uses the character indexes to identify a given span of text. In the programming language Python, for example, assuming that the variable `document_text` is a list of characters representing the entire text of the document APh 75–0113, the notation `document_text[442:453]` selects the string of text corresponding to “Pliny, nat.”.

T1	REFAUWORK	442	453	Pliny, nat.
T2	REFSCOPE	454	463	11, 4, 11
T3	REFSCOPE	464	478	and 11, 16, 46
T4	REFAUWORK	483	497	Vergil, georg.
T5	REFSCOPE	498	509	4, 149-218.

Figure 4.7: The annotation of a canonical reference represented as standoff markup. The four columns contain respectively: the annotation identifier; the named entity label; the start and end index of the annotated string; the annotated portion of text.

The reason for keeping two distinct representations of the same data lies in the fact that different formats suit different tasks in different ways. Similarly, the decision about which formats to use is determined by the pieces of software or libraries used. In this specific case, the choice of using both the IOB and the standoff markup format was due to the fact that a tabular format such as IOB is an input format commonly accepted by NLP tools, whereas the latter is the format that Brat uses internally to store the annotated data. To keep these two representations of the same underlying data synchronised I had to carry out some additional operations that need careful testing as they may lead to the propagation of errors throughout the corpus.

4.4 THE INFORMATION EXTRACTION PIPELINE

In this section I provide a detailed description of the information extraction pipeline that I have implemented in order to capture canonical citations from the datasets that were just described.²⁰ The term *pipeline* emphasises the fact that the processing steps form a sequence where the output of the previously executed step constitutes the input for the following one (see figure 4.8).

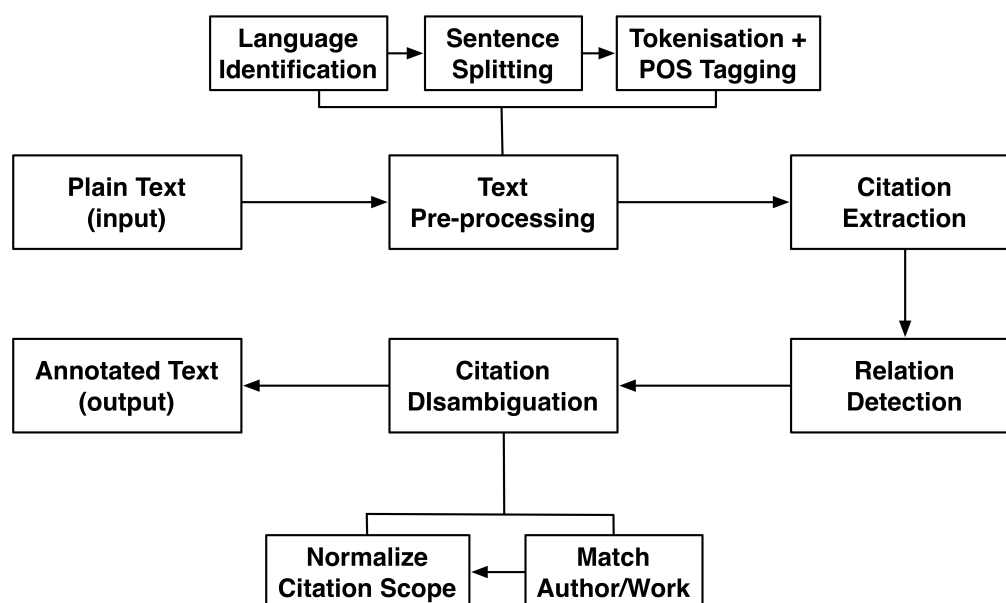


Figure 4.8: Overview of the information extraction pipeline.

The pipeline takes as input a plain text document such as the one in figure 4.9 and returns as output a list of items like the one given in table 4.5. Each item consists of a label, a type – as defined in my annotation scheme (section 4.3.1) – and a unique identifier, expressed as a CTS URN, which links the item to the corresponding record in the knowledge base.

In this section I describe how I implemented each of the intermediate steps that are needed to transform a plain text into the desired output. I also touch upon the challenges that I faced during the development process, but this point is articulated more fully in section 4.5 where I discuss the evaluation of the information extraction pipeline.

²⁰ My implementation of this pipeline consists of two components – the citation-extractor and the citation-parser – which are both available under an open source licence respectively at <http://dx.doi.org/10.5281/zenodo.35470> and <http://dx.doi.org/10.5281/zenodo.35472>.

Text of APh 75-06176 (named entities highlighted in bold)

Vergil has been appropriated for many different perspectives on the world. These appropriations involve assimilations that tend to erase the particularly Roman aspects of **Vergil**. It is important to make, or keep, Vergil strange, especially in the area of translation. A defamiliarizing reading of **Aen. 1,1-11** helps to illustrate how some English translations of the « **Aeneid** » map onto the spectrum of assimilation-dissimilation.

Figure 4.9: An example of the pipeline input: the text of APh 75-06176.

Table 4.5: An example of the pipeline output: for each named entity or relation contained in APh 75-06176 the table shows the corresponding string, the annotation type and the identifier that disambiguates it.

String	Type	Identifier (CTS URN)
Vergil	Aauthor	urn:cts:latinLit:phi0690
Vergil	Aauthor	urn:cts:latinLit:phi0690
Aen. 1,1-11	Scope	urn:cts:latinLit:phi0690.phi003:1.1-1.11
« Aeneid »	Awork	urn:cts:latinLit:phi0690.phi003

4.4.1 Pre-processing

The pre-processing phase entails a number of operations that are necessary in order to transform an unstructured stream of text into a more structured collection of sentences and tokens. Two examples of information that are added to the input text at this stage are the language in which a given text is written and the lexical syntactic category for each token in the text (i.e. PoS tag).

Language Identification

The grammatical categories – or PoS tags – that are assigned to a token vary depending on its language. Therefore, it is necessary to identify the language of a text first in order to extract the appropriate set of PoS tags for all the tokens it contains.

The language identification was performed by using `guess_language`²¹, a Python library that is able to recognise over 60 languages. The library uses the frequency of trigrams (i.e. sequences of three characters) in order to determine the language of the input text. Although the accuracy of this library was not the object of formal evaluation in this study, close

²¹ Guess-language 0.2, <https://pypi.python.org/pypi/guess-language>.

inspection of the results revealed that in the vast majority of cases the library's guess was accurate.

Sentence Segmentation

The main reason why texts are split into sentences when extracting canonical citations is that the co-occurrence of two or more citations within the same sentence may be an important indicator of their relatedness. In this sense, the sentence constitutes a unity of context that is granular enough to be meaningful, especially when the text is fairly long and contains many citations of primary sources such as in journal articles. In fact, it does make a difference to say that two citations are related because they occur within the same sentence as opposed to being found within the same article.

The automatic splitting of a text into sentences is usually a straightforward task. However, the high density of abbreviations that characterise publications in domains such as Classics constitute a challenge for many NLP tools that are available for this purpose.

When a tool does not have a robust enough way of handling abbreviations, the trailing dot of an abbreviation is mistakenly interpreted as a final stop signalling the end of a sentence. Since the sentence segmentation is typically performed at the beginning of a pipeline, an error at this stage may initiate a cascade of errors in the subsequent steps, thus leading to a drastic deterioration of the overall performance of the system.²²

The comparison of the different solutions that I have tested showed that the most reliable way of splitting a text containing a high number of abbreviations into sentences was to use the sentence segmentation script provided by Brat.²³ This fact, however, is hardly surprising given that Brat has been developed by the BioNLP community where biomedical texts like publications in Classics are characterised by an extensive use of abbreviations.

²² See *infra* p. 162 for an example of this phenomenon.

²³ The Python script `sentencesplit.py` can be found at <https://github.com/nlplab/brat/blob/master/tools/sentencesplit.py> and is based on the sentence splitter that was used to create the GENIA corpus, an annotated corpus of biomedical abstracts. The source code for the GENIA sentence splitter can be found at <https://github.com/ninjin/geniass/>.

Tokenisation and PoS Tagging

Tokenisation is the process of breaking up a stream of text into smaller chunks that are called tokens, while PoS tagging consists of assigning to each token the corresponding lexical syntactic category (e.g. verb, adverb, pronoun, etc.). As previously mentioned, the set of PoS tags varies depending on the language and for some languages like English several competing tagsets exist.

It is worth noting that tokens do not necessarily always correspond to words: for instance, when tokenising the string “influenced by Pliny, nat. 11, 4, 11 and 11, 16, 46 and” (see figure 4.6) each punctuation sign becomes a separate token. For this reason, and especially when referring to the size of the training set, it is more appropriate to refer to tokens rather than words as atomic units.

The main argument for extracting PoS tags is that they proved to be one of the most informative features for the extraction of named entities from text (Romanello, 2013, p. 13). To perform both the tokenisation and PoS tagging I used the tool TreeTagger²⁴, which is a probabilistic tool that supports a number of languages including English, French, German, Italian and Spanish (Schmid, 1994,9). Several interfaces to this tool have been written in different programming languages and I used one written in Python²⁵.

Abbreviations constituted a challenge for the text tokenisation as they did for the sentence segmentation. The string “nat.”, for example, would be erroneously split into two separate tokens: “nat” tagged as being a proper noun and “.” with PoS tag “SENT”, which stands for end punctuation. In order to overcome this issue, TreeTagger offers the possibility of specifying a list of abbreviations to use when tokenising a text. Since the most common abbreviations for author names and work titles are included in the knowledge base, a list of such abbreviations can easily be generated and passed to TreeTagger.

4.4.2 Named Entity Extraction

The extraction of named entities from text is the first real processing step after the preparatory steps that were just discussed (i.e. language

²⁴ TreeTagger, <http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger>.

²⁵ treetagger-python, <https://github.com/miotto/treetagger-python>.

detection, sentence segmentation, tokenisation and PoS tagging). The entities to be captured, as described in section 4.3.1, are Aauthor, Awork, Refauwork and Refscope. Various combinations of these four entities allow us to annotate virtually any kind of canonical citation, from the more concise and structured to the more discursive.

The approach that I have adopted to tackle this task is based on machine learning and specifically on supervised learning algorithms, meaning that the rules for the identification of these entities are not specified in advance but are extracted from a training set and learned by a statistical model.²⁶ Assigning named entity tags to tokens is decided by the model based on a set of features that is computed for each token (i.e. the feature vector). This section describes the feature set I have devised for the extraction of citations and other relevant named entities. In section 4.5.1 I compare the performance of different statistical models trained using the same feature set and the same training data.

Linguistic Features

Since the system was designed to be as language-independent as possible, the number of linguistic features to extract has been kept to a minimum as linguistic features are always language-dependent. First, the PoS tag as extracted by TreeTagger was included in the feature set without performing any manual correction. Second, the neighbouring words of each token w_i in the range $w_{i-2} \dots w_{i+2}$ were considered as features. Interestingly, experiments with using features such as word suffixes and prefixes of up to 4 characters in length in addition to the neighbouring tokens showed a degradation in the performance.

Word-level Features

Additional features were captured describing different aspects of the characters that form a token (e.g. punctuation, case, etc.). As the evaluation of the performance showed, this kind of features alone are responsible for a substantial improvement (i.e. +18.37%) of the overall accuracy of the system (Romanello, 2013, p. 13). These features are modelled after those proposed by Councill et al. (2008) for the extraction of modern bibliographic references and were designed specifically

²⁶ The algorithms that were used, as well as the reasons for choosing them, are presented in section 4.5.1.

to leverage the characteristics of canonical citations. For example, the punctuation and case features capture patterns of capitalisation as well as the presence of characters such as hyphens, quotation marks and brackets that are often found within or in conjunction with canonical citations (see tables 4.6 and 4.7).

Table 4.6: Named entity extraction: selected examples of punctuation features.

Feature Value	Example
final_dot	"georg."
continuing_punctuation	"," " "
quotation_mark	"«", "»"
has_hyphen	"61-73"
bracket	"(", "]"
no_punctuation	"view"

Table 4.7: Named entity extraction: selected examples of case features.

Feature Value	Example
mixed_caps	"APh"
all_caps	"BS"
init_caps	"Verg."
all_lower	"luogo"

Table 4.8: Named entity extraction: selected examples of number features.

Feature Value	Example
number	"322"
roman_number	"XI"
Year	"1900"
range	"13-19"
dot_separated_number	"5.1.10"
dot_separated_plus_range	"1.1-1.10"
mixed_alphanum	"12a"
no_digits	"those"

Moreover, a set of features was required to capture the characteristics of the Refscope entities – i.e. the entities that represent the scope of canonical citations. Such entities constitute approximately one-third of the total number of entities in the training set and are characterised primarily by the presence of numbers and punctuation signs. For these

reasons, the number feature covers a wide variety of aspects that are commonly found in the scope of citations (see table 4.8): a) the use of Roman numerals to indicate the book number of the cited work; b) the use of the hyphen for citations that refer to a range of text sections; c) the use of the dot “.” to distinguish the hierarchical levels of the cited work and d) the presence of a mixture of digits and letters that characterises citations adhering to specific citation schemes such as Bekker numbers or Stephanus.²⁷

Finally, the pattern feature aims to create relations between words that are not identical but that exhibit similar patterns concerning the sequence of characters (table 4.9). The first of these patterns, the so-called *extended pattern*, is computed by replacing lowercase characters with “a”, uppercase ones with “A”, numbers with “o” and punctuation signs with “-”. For example, the extended pattern of “Avien.”, which stands for Avienus, is “Aaaaa-” and is identical to the pattern of “Strab.”, which indicates the author Strabo. Additionally, a compressed pattern is extracted from each token by replacing sequences of similar characters with one single pattern character: “Aaaaa-” is compressed into “Aa-” and, as a result, the abbreviations “Avien.”, “Strab.” and “Thuc” (and many others) share the same compressed pattern (“Aa-”). These patterns aim to capture high-level similarity between strings.

Table 4.9: Named entity extraction: selected examples of pattern features.

Feature Value	Example
extended pattern	“Avien.” → “Aaaaa-”
compressed pattern	“Avien.” → “Aa-”

Semantic Features

Semantic features – or list lookup features – unlike the features that have been considered so far capture some aspects of the meaning of strings. Four different semantic features are extracted from each token and indicate whether or not the target token matches a pre-compiled list containing names, titles and abbreviations (see table 4.10). This list, often referred to as a dictionary or gazetteer, is drawn from the ontological knowledge base, as is the list of abbreviations used to improve the accuracy of tokenisation and sentence segmentation. The semantic

²⁷ For a discussion of these citation schemes see *infra* at p. 58.

features distinguish between words that partly match a name or title (e.g. “women” in “The catalog of women”) from words that produce a total match (e.g. “Aristofane” or “Theogony”). This distinction is particularly useful when handling author names or work titles that consist of more than one token.

Table 4.10: Named entity extraction: selected examples of semantic features.

Feature Value	Example
match_authors_dict	“Aristofane”
contained_authors_dict	“of”
match_works_dict	“Theogony”
contained_works_dict	“women”

At this point of the pipeline the language of the input text was automatically detected and the text itself split into sentences. In turn, each sentence was broken up into tokens that were automatically tagged with their lexical syntactic category and, finally, authors, works and other citation components were automatically identified.

4.4.3 Relation Detection

The next step to be performed is to detect the relations existing between named entities. As previously illustrated, the annotation scheme contains the scope relation which represents a citation as a relationship between two entities. The input is the text annotated with named entity information and the output is a list of relations, where each relation represents a canonical citation; in turn, each relation consists of a relation type (i.e. scope) and the named entities that are involved in the relation, which are also called the arguments of the relation. By definition, the second argument of the relation is always a Refscope entity, while the first argument can have type Aauthor, Awork or Refauwork. As a result, there are three possible combinations of named entities that can be part of a scope relation:

1. arg1=Aauthor, arg2=Refscope; e.g. “Ammianus (15, 8, 7)”;
2. arg1=Awork, arg2=Refscope; e.g. “Trabajos 159–173”;
3. arg1=Refauwork, arg2=Refscope; e.g. “Pliny, nat. 11, 4, 11”.

Rule-based Detection

To determine when a relation between any two named entities in a text exists I have devised a set of rules expressed by algorithm 1.

Algorithm 1 Pseudocode for the FindRelations function

```

1: procedure FINDRELATIONS(entities)
2:   relations  $\leftarrow$  list()
3:   for all entity  $\in$  entities do
4:     if entity.type  $\neq$  "Refscope" then
5:       arg1  $\leftarrow$  entity
6:     else if entity.type = "Refscope" then
7:       if arg1  $\neq$  None then
8:         arg2  $\leftarrow$  entity
9:         relations  $\leftarrow$  list(arg1, arg2)
10:      end if
11:    end if
12:  end for
13:  return relations
14: end procedure

```

The design of the algorithm is based on the observation that the cited passage normally follows the mention of the cited author or work and is applied sequentially from left to right to all named entities in the input text (1.3). The relations are stored in a list, which is empty at the beginning of the procedure (1.2). Whenever an entity with type Aauthor, Awork or Refauwork is encountered (1.4) this entity is retained as a possible relation candidate and thus stored in the variable arg1 (1.5). If this variable already contains a previously found entity, its value is replaced with the current entity. Moreover, whenever a Refscope entity is found (1.6), the algorithm first checks if a relation candidate was already found (1.7): if this is the case, the current entity is assigned to the variable arg2 and a new relation is created and added to the list (1.9); otherwise, nothing happens and the algorithm moves on to the next entity.

Figure 4.10 illustrates how the algorithm works by representing schematically the status of a programme that implements this algorithm at four different execution steps, each of them corresponding to one iteration of the loop at the lines 1.4-1.12. At step 1 the entity being considered is "Ammianus": since its entity type is Aauthor (1.4), this entity is assigned to the variable arg1 (1.5) and the algorithm moves on the next entity. The steps 2 and 3 are identical to step 1, with the only difference

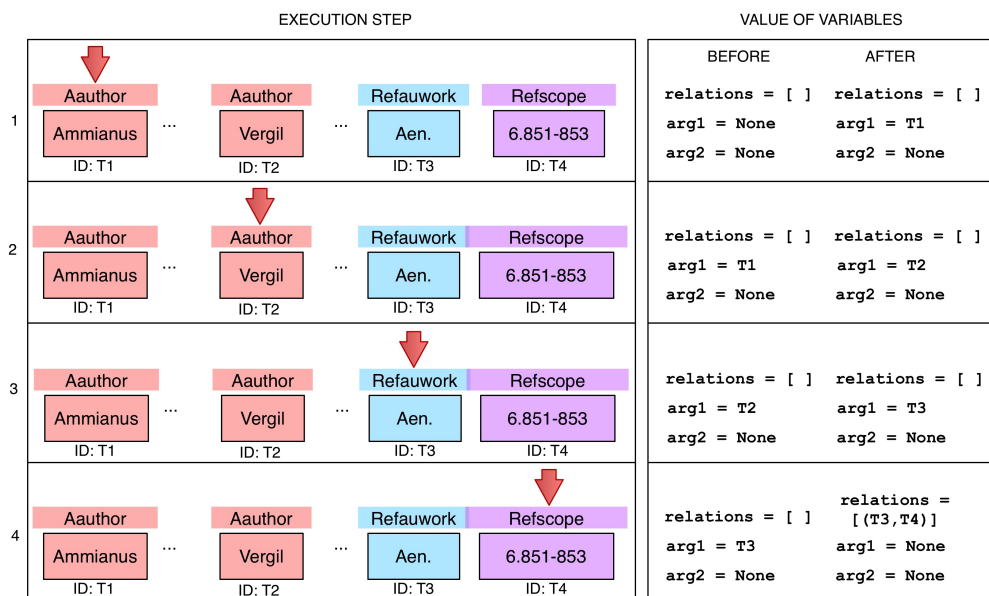


Figure 4.10: This diagram depicts the execution of the algorithm over four entities: the tokens occurring between these entities are not represented as the algorithm considers only named entities. The entity that is being processed at each step is indicated by a red arrow; on the right hand side the value of the variables before and after executing that step is provided.

that the value of variable `arg1` changes. When the algorithm reaches step 4, the variable `arg1` points to “Aen.”; since the entity type now in focus is Refscope (1.6) the entity is assigned to the variable `arg2` (1.8) and a new relation is created and added to the list of relations (1.9).

This algorithm enables the identification of relations that may extend across multiple sentences as well as consecutive citations that are constituted by non consecutive tokens (for an example see figure 4.4, p. 124). However, since this algorithm is designed to process the text from left to right, it is not suitable for capturing the more discursive citations such as “le livre 18 de la « Chronographia »” where the citation scope occurs before the reference to the cited work.²⁸

4.4.4 Entity and Relation Disambiguation

This step of the information extraction pipeline is directly concerned with the content or meaning of the pieces of information that are cap-

²⁸ For a discussion of the impact of this kind of citations on the evaluation results see *infra* p. 155.

tured. The goal of this step is to identify unambiguously each extracted entity mention and relation (i.e. citation) by means of a unique identifier.

The following example summarises what is the result of the extraction steps examined up to this point. The article entitled *Propertiana* that was published in 2004 by W. S. Watt in the journal *Rheinisches Museum für Philologie* is summarised by the APh abstract 75–04686, which reads as follows (named entities highlighted in bold):

Textkritisches zu **Propertz 1, 2, 9–14 ; 1, 18, 25–28 ; 2, 6, 31–32 ; 2, 25, 21–22 ; 2, 32, 23–24 ; 3, 7, 57–60 ; 3, 21, 31–32 und 4, 3, 7–10.**

Albeit concise, the abstract is highly informative as it allows the reader with an interest in textual criticism to know what specific passages of Propertius' *Elegiae* are debated in this article. After performing named entity extraction and relation detection on this text, a list of canonical citations is obtained and represented as relations between the various named entities that constitute the citation:

R1 Arg1: ("T1", "Refauwork", "Propertz") Arg2: ("T2", "Refscope", "1, 2, 9-14 ;")
 R2 Arg1: ("T1", "Refauwork", "Propertz") Arg2: ("T3", "Refscope", "1, 18, 25-28 ;")
 R3 Arg1: ("T1", "Refauwork", "Propertz") Arg2: ("T4", "Refscope", "2, 6, 31-32 ;")
 R4 Arg1: ("T1", "Refauwork", "Propertz") Arg2: ("T5", "Refscope", "2, 25, 21-22 ;")
 R5 Arg1: ("T1", "Refauwork", "Propertz") Arg2: ("T6", "Refscope", "2, 32, 23-24 ;")
 R6 Arg1: ("T1", "Refauwork", "Propertz") Arg2: ("T7", "Refscope", "3, 7, 57-60 ;")
 R7 Arg1: ("T1", "Refauwork", "Propertz") Arg2: ("T8", "Refscope", "3, 21, 31-32")
 R8 Arg1: ("T1", "Refauwork", "Propertz") Arg2: ("T8", "Refscope", "4, 3, 7-10.")

The final step is to associate each citation with the corresponding unique identifier, expressing in a machine readable format which text passage is being cited. For example, the canonical reference "Propertz 1, 2, 9–14" – one of the *loci* addressed by Watt's article – needs to be transformed into the CTS URN urn:cts:latinLit:phi0620.phi001:1.2.9-1.2.14. Applying this to the list above results in the following list of identifiers:

R1 urn:cts:latinLit:phi0620.phi001:1.2.9-1.2.14
 R2 urn:cts:latinLit:phi0620.phi001:1.18.25-1.18.28
 R3 urn:cts:latinLit:phi0620.phi001:2.6.31-2.6.32
 R4 urn:cts:latinLit:phi0620.phi001:2.25.21-2.25.22
 R5 urn:cts:latinLit:phi0620.phi001:2.32.23-2.32.24

R6 urn:cts:latinLit:phi0620.phi001:3.7.57-3.7.60
 R7 urn:cts:latinLit:phi0620.phi001:3.21.31-3.21.32
 R8 urn:cts:latinLit:phi0620.phi001:4.3.7-4.3.10

The operation of transforming the human readable notation “Properz 1, 2, 9–14” of the example above into a machine actionable identifier entails two distinct sub-steps. First, it is necessary to determine that the string “Properz” is commonly used within the context of a citation as a shortcut to refer to the *Elegies*, identified by the URN urn:cts:latinLit:phi0620.-phi001. Second, it is necessary to map the citation scope “1, 2, 9–14” – which translates to “book 1, poem 2, lines 9 to 14” – to the normalised form “1.2.9–1.2.14”. In such a normalised form compressed ranges of passages are expanded and hierarchical levels of the cited work are dot-separated.

Assigning each citation the correspondent unique identifier requires a substantial amount of information that is not contained in or can be deduced from the text. This sort of information is contained in the knowledge base about the domain of Classics discussed in chapter 3. This knowledge base plays the most central role in the disambiguation step as it holds information such as unique identifiers, name or title variants and information about the *opus maximum* of a given author.

Matching Entities against the Knowledge-Base

Generally speaking, matching the extracted Aauthor and Awork entities against the knowledge base consists of looking up the string to match in a set of dictionaries. These dictionaries, as already mentioned, consist of abbreviations and variants of names of ancient authors and titles of works. Some clean-up and normalisation is performed on the search string prior to the lookup: punctuation signs that may surround or be included in the extracted named entity are removed and the string is transformed to lowercase in order to facilitate the matching. Similarly, all names and titles are turned to lowercase when constructing the dictionaries; additionally, articles that are present within work titles are removed (e.g. *The Acharnians* becomes “acharnians”). Without removing the articles from the titles, a lookup of “Acharnians” would return no exact matches as the normative titles contained in the knowledge base always comprise the determinative article.

Matching Refauwork entities is slightly more complicated as this entity type captures strings that need to be treated in different ways. Strings

like “Pliny, nat.”, “Georg.” or “Thuc.” are all labelled by the system as Refauwork. As a result, it is necessary to apply a set of heuristics when matching entities of this kind in order to determine the dictionary in which the string should be looked up. First, the programme assumes that the string is a title (or the abbreviation of a title) and tries to match it against the dictionary of titles. If no matches are found, the programme looks up the string against the dictionary of author names. Finally, if the second attempt does not succeed and provided that the string contains more than one word, the programme tries to split it and look up the words that constitute the string in separate dictionaries.

So far I have talked about *matching* without further specifying what type of matching is being performed. Matching of two strings can either be exact or approximate: *exact matching* means that two strings match when they are identical, whereas *approximate matching* matches strings that are similar. The similarity between strings can be quantified by using several metrics. The metric I used in my research is known as Levenshtein distance or simply as *edit distance* and measures the dissimilarity between two strings in terms of operations – i.e. insertions, deletions and substitutions – that need to be performed in order to transform one string into the other. For example, the string “Vergil” and “Virgilio” have an edit distance of 3: in fact, to transform the former into the latter one needs to substitute “e” with “i”, insert an “i” after “Virgil” and append an “o” at the end. Since each of these operations has a cost of 1, the distance equals the sum of the costs, i.e. 3.

When performing approximate matching it is useful to specify a threshold so that matching candidates with an edit distance that exceeds such a threshold can be discarded. This type of matching, as opposed to exact matching, may be a desirable approach when trying to match a string that may be wrongly transcribed, for instance due to OCR errors. In the specific case of matching names and titles, approximate matching within a certain threshold proved to be a viable solution for me given the nature of the data I have been dealing with. In fact, variants in different languages of the name of an ancient author tend to have a relatively low edit distance as they generally stem from the Latin or Greek name (e.g. “Homer”, “Homère”, “Homerus” and “Omero”). Therefore, this approach is useful when the string to be matched is not contained

in the knowledge base but partly matches against other author names contained in it.²⁹

Normalisation of Citation Scope

The final step of the pipeline is to normalise the part of a canonical citation that indicates which specific part of a given text is being cited, the so-called *citation scope*. Since the same scope may be indicated by means of several equivalent representations it becomes necessary to map all such variants to a common normalised form in order to minimise variation and ambiguity. For example, a reference to Thucydides' *Histories* book 1, chapter 89, sections 1–2 may be written as:

1. Thuc. 1.89.1–2
2. Thuc. 1, 89, 1–2
3. Thuc. I 89, 1s.

Transforming these human-readable notations into a computationally tractable representation requires absolute explicitness and consistency (see section 3.1). Therefore, the citation scope needs to be mapped onto a normalised representation, in this case “1.89.1–1.89.2”. The human reader, however, will interpret these three notations in the same way, despite the fact that they are expressed in slightly different ways. The first two citations differ only by the punctuation symbol used to separate the hierarchical levels of the cited work (i.e. book/chapter/section) respectively the dot “.” in the former and the comma “,” in the latter. The third citation uses the Latin ordinal number to indicate the book and the abbreviation “s.” – which in Italian and French stands respectively for *segunte* and *suivant* (i.e. “following”) – to refer to the following section. Moreover, when the citation scope consists of a range of passages some details may be left implicit to avoid repetitions, thus leaving to the reader the task of figuring out the omitted pieces of information. In the examples above, for instance, the range of passages may be expressed as “1.89.1–2”, instead of the more verbose form “1.89.1–1.89.2”, because the cited sections are both found in the same book and chapter.

²⁹ In my research I have experimented with both exact and approximate matching. In section 4.5.3 I discuss the respective drawbacks and advantages of using these two approaches.

The technique I have used in order to process and normalise the citation scopes is called *parsing*, and specifically parsing based on Context-free Grammar (CFG). As part of my research I built a parser that follows a set of rules specified as a CFG in order to extract some structure from citation scopes.

A good example of the purposes for which CFGs are widely employed is the compiling of programming languages, namely the transformation of code into a set of low-level instructions that can be executed by the machine. Each programming language has its syntactic rules that any code written in that language must follow and these rules can be expressed by means of a CFG. By relying on these rules, the compiler can check if any piece of code is well-formed or, in other words, if all the syntactic rules of the language in which the code is written are followed correctly. Typically, a compiler uses a parser in order to process the code being compiled and in doing so relies on a grammar of rules that describe the syntax of that specific language.

CFGs can be used to parse specific kinds of strings such as patent numbers or citation scopes because they resemble closely a formal notation that follows a finite set of rules. Recently, CFGs have been successfully applied to the processing of specific sections of scholarly publications in Classics such as indexes and critical apparatuses, which follow a relatively rigid structure. Boschetti (2007), for example, has applied CFG-based parsing to critical apparatuses based on the observation that their structure is suitable for formalisation by means of a grammar.³⁰ Similarly, I have demonstrated elsewhere that parsing can be successfully applied to capture the structure of indexes of cited passages that can be found at the end of scholarly publications (Romanello et al., 2009a).

The framework I have used to implement the parser is called ANother Tool for Language Recognition (ANTLR)³¹, which is an open source framework for building parsers (Parr and Quong, 1995). Although ANTLR is mostly used to build grammars for the parsing of programming languages, it has also been applied to NLP problems such as for example the extraction of patent numbers from legal texts (Surdeanu et al., 2014).

³⁰ The critical apparatus is the section of a critical edition where the editor records variant readings and conjectures about the text and where references backing the editor's choices in establishing the text are provided.

³¹ ANTLR, <http://www.antlr.org/>.

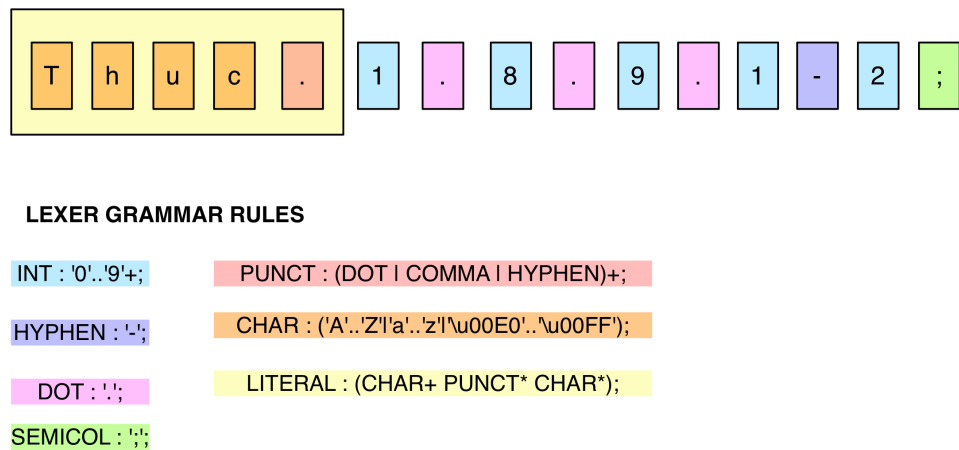


Figure 4.11: Schematic representation of the lexer grammar rules that are matched when tokenising the string “Thuc. 1.89.1-2;”; each token is coloured according to the rule it satisfies.

Since providing a detailed explanation of how the parser was implemented and how CFGs work falls outside the scope of this dissertation, I shall attempt to illustrate the underlying idea by describing how the citation “Thuc. 1.89.1-2;” can be parsed by using a CFG-based parser (figures 4.11 to 4.13).

First of all, the ANTLR parser consists of three components: a lexer, a parser and a tree parser. Each of these components is described by a grammar of rules which are usually applied sequentially. The lexer grammar defines a set of rules for splitting the text stream into a sequence of smaller blocks, the tokens. Figure 4.11 shows schematically the set of tokenisation rules that was defined and the result once they are applied to the string “Thuc. 1.89.1-2;”. A character, represented by the token CHAR, is defined as any character in the range “a-z” and “A-Z”. Token rules can also be combined to form more complex building blocks: the token LITERAL, for example, is defined as (CHAR+ PUNCT* CHAR*), which means a sequence of one or more characters that may contain also one or more punctuation signs (e.g. “Thuc.”).

The stream of tokens that is outputted by the lexer becomes the input for the parser (figure 4.12). The parser rules describe in a recursive way which sequences of tokens are expected, and thus accepted, by the parser. The top-level rule ((ref (ref_separator ref)*)) says that the input string is a sequence of one or more references separated by a semicolon. A reference, identified by the rule ref: work* scope, is

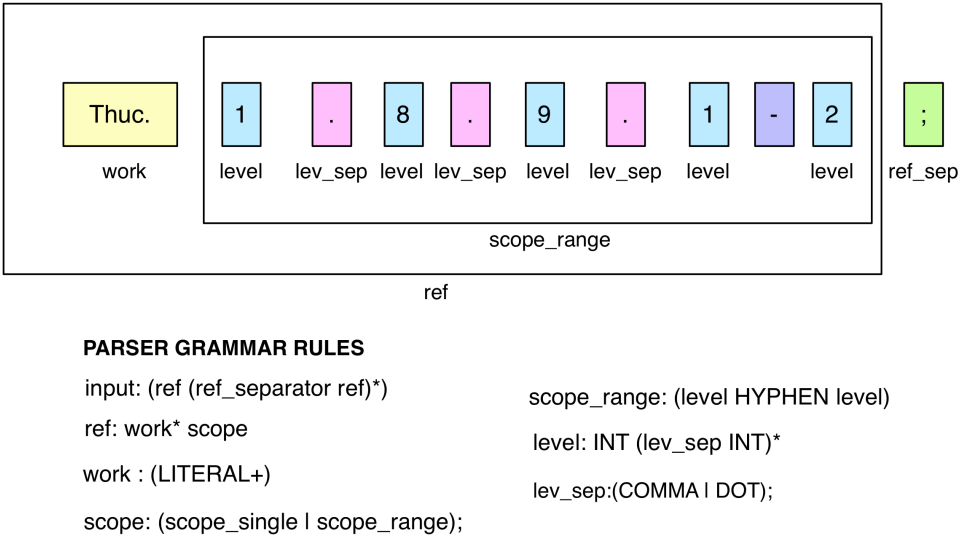


Figure 4.12: Schematic representation of how the stream of tokens representing the string “Thuc. 1.89.1-2;” is parsed according to a set of grammar rules (only the rules that are matched are displayed).

further defined as a sequence of work and scope elements; the work element may be omitted, as indicated by the star “*” symbol, meaning that both “Thuc. 1.89.1-2;” and “1.89.1-2;” are valid inputs for this parser. The grammar rules continue with more fine-grained rules such as level and lev_sep.

The output of this parser is a parse tree, namely a tree structure that represents the hierarchy of elements that are produced by a grammar of rules (see figure 4.13a). Such a parse tree can then be traversed by using a tree parser and transformed into a data structure such as the fragment of JavaScript Object Notation (JSON) depicted in figure 4.13b. While traversing the tree some operations are performed such as making fully explicit the compressed notation used to refer to contiguous passages of text (e.g. “1.89.1-2” is transformed into “1.89.1-1.89.2”).

4.5 EVALUATION OF THE PERFORMANCE

An important aspect of automating the extraction of canonical citations is the ability to measure the accuracy with which the task is carried out. In this case the accuracy refers to how many citations were correctly identified and how many were missed. In this section I present an evaluation of the system to extract such references described in the previous

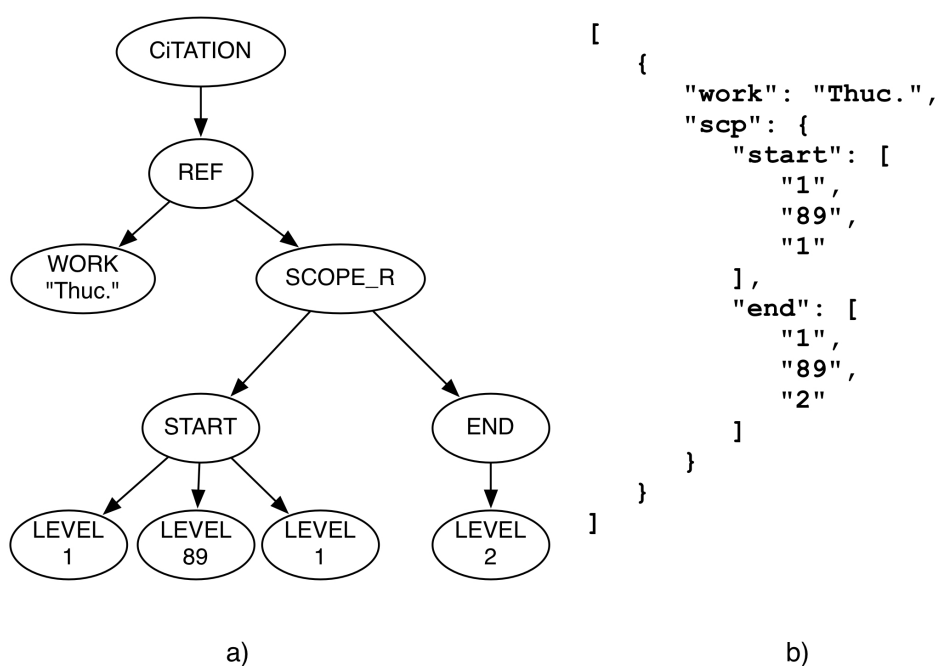


Figure 4.13: This figure shows a) the parse tree resulting from parsing the string "Thuc. 1.89.1-2;" and b) a JSON representation of the parse tree after normalising the range of cited passages.

section.³² The three steps that make up the evaluation process will now be considered separately:

1. **The extraction of named entities:** the identification of names, titles and references in the text;
2. **The detection of the relations that exist between entities:** since a reference is represented as a relation between two entities, the canonical references are reconstructed starting from the entities that are found in the text;
3. **The disambiguation of named entities and relations:** the system tries to determine which entity, from those that are contained in the knowledge base, is referred to in the text; in order to do so each entity and relation is assigned a unique identifier, a CTS URN.

³² The evaluation results discussed later in this section were obtained by using version 1.3.4 of the citation-extractor <http://dx.doi.org/10.5281/zenodo.35470>, version 0.4.0 of the citation-parser <http://dx.doi.org/10.5281/zenodo.35472> and version 1.0 of the APh dataset <http://dx.doi.org/10.5281/zenodo.35471>.

The kind of evaluation to be carried out depends on the nature of the solution being evaluated. When evaluating machine learning-based solutions it is necessary to keep separate the training and the test set. For the evaluation to be meaningful, the data that was used to train the system must not be used in the evaluation. Obviously, this necessity does not apply to the evaluation of rule-based algorithms as the classification rules are defined a priori and are not derived from observations on the data.

Among the methods that are used in statistics for evaluation purposes I have employed the *k-fold cross-validation*, specifically the 10-fold cross-validation, a method that is commonly used for the evaluation of NLP systems. A key concept of cross-validation is that the dataset is divided into k subsets and the evaluation is performed on each subset in successive *rounds*. To perform a 10-fold cross-validation the dataset is divided into 10 subsets: at round 1, the first subset is used for training and the remaining (9) subsets are used for evaluation, i.e. to compute precision, recall and F_1 score. This procedure is repeated for the remaining subsets and the evaluation results are then averaged over the rounds.³³

The data used for the evaluation was drawn solely from the APh corpus (section 4.3.2) as it was the only one to be manually corrected. The metrics that are used to assess the performances of the system – precision, recall and F_1 score – were introduced in section 4.5.

It is worth noting that one limit of this evaluation is the lack of an already existing baseline with which the results can be compared. To overcome this limitation, which is due to the innovative nature of my research, I have compared the results of using different solutions to perform the same task (e.g. the use of different statistical models for the extraction of named entities).

4.5.1 *Evaluation of Named Entity Extraction*

The first step in extracting canonical references from text is the automatic identification of author names, work titles and references to specific work passages as these entities constitute the building blocks of such references. These blocks are mapped onto named entities – i.e. Aau-

³³ It should also be noted that the implementation of 10-fold cross-validation that I have used balances the number of positive and negative instances that are contained in each subset.

thor, Awork, Refauwork and Refscope – as was described in section 4.3.1. Since an entity may consist of multiple tokens (e.g. the title “Works and Days”), each entity is in fact represented by two distinct tags: the first token of an entity is tagged with the name of the entity itself to which the prefix B-, which stands for beginning, is appended; for all the following tokens the prefix I- is used.³⁴ For example, the title *Works and Days* is represented as:

B-AWORK/Works I-AWORK/and I-AWORK/Days

As a result, extracting named entities is the task of classifying a sequence of tokens by assigning to each token one of the following tags: B-AUTHOR, I-AUTHOR, B-AWORK, I-AWORK, B-REFAUWORK, I-REFAUWORK, B-REFSCOPE, I-REFSCOPE or the tag 0 if the token does not constitute a named entity.

What was evaluated is how well different machine learning algorithms perform the task of extracting named entities from text. The dataset that was employed for the 10-fold cross-evaluation contains in total 25,889 tokens and 1,261 named entities. The performances are assessed based on the standard measures of precision, recall and F_1 score. However, one distinction needs to be made concerning how these measures were computed. I did consider the number of correct *tags* whereas in other contexts, such as the CoNLL shared task, the calculation is based on the number of correct *entities*. The difference between the two approaches is that if the system outputs the sequence 0/I-AWORK/I-AWORK instead of B-AWORK/I-AWORK/I-AWORK this counts as two true positives and one false negative, instead of considering the entire entity as not correctly recognised.

Three machine learning algorithms have been evaluated on this task: Conditional Random Fields (CRF), Support Vector Machines (SVM) and Maximum Entropy (MaxEnt).³⁵ These algorithms are commonly applied to the task of NER as well as to other tasks. The CRF model, which was theorised by Lafferty et al. (2001), has been applied to a wide range of classification problems including computer vision and bioinformatics and is currently considered the state-of-the-art method in sequence labeling tasks such as NER.³⁶ The SVM was theorised by

³⁴ This file format was explained in section 4.3.3.

³⁵ In writing this section I have used as references Jurafsky and Martin (2009), Manning et al. (2008) and Hastie et al. (2009).

³⁶ For an introduction to CRF and its possible applications see Sutton and McCallum (2006).

Vapnik (1995) and ever since has been successfully and widely used in text classification and sequence labelling applications, including NER (see e.g. Mayfield et al., 2003). Finally MaxEnt, also called Multinomial Logistic Regression model, has been applied since the 1990s to many supervised problems in NLP such as for example sentence-boundary detection and machine translation. Applications of MaxEnt to NER are e.g. Borthwick et al. (1998) and Chieu and Ng (2003).

Discussing in detail the foundations of these algorithms falls outside the scope of this dissertation as it would require introducing a number of very technical concepts from mathematics and statistics. However, it is worth making a few general remarks concerning the nature of these algorithms.

First, CRF, SVM and MaxEnt are all supervised learning algorithms in the sense that they take as input labelled data and use the input to predict the value of the outputs, yet they use different methods to predict the output. Two of the chosen algorithms – CRF and MaxEnt – are probabilistic in that the decision of which class label needs to be assigned to a given token is based on the probability of assigning that class given the observed features. SVM, on the other hand, represents each instance as a multidimensional vector of features and uses the vector space to classify such instances.³⁷

Second, only CRF is optimised for labelling sequences of items rather than single instances, what in machine learning terms is called the prediction of structured outputs. There are two possible solutions to obviate this issue so that the comparison between different algorithms remains meaningful. The first solution is to use combinations of these algorithms with Hidden Markov Models (HMM), such as SVM^{hmm} and MaxEnt Markov Models, which are optimised for sequence labelling tasks. The second solution – the one I have adopted here – is to employ a sliding-window labeller which allows the classifier to take into account the features assigned to the preceding and following tokens within a window of a certain size for training and classification purposes (in this specific case, the two preceding and two following tokens are considered).

The evaluation results I shall present next were produced by using existing implementations of these three algorithms, specifically a C++ im-

³⁷ For a detailed description of the feature set see section 4.4.2.

plementation of CRF, called CRF++³⁸ and the SVM and MaxEnt implementations that are provided by scikit-learn³⁹ (Pedregosa et al., 2011), a machine learning library written in Python. Moreover, the statistical models were employed without modifying the default values of the parameters that each model implementation provides.⁴⁰

The CRF method produced the best overall results with overall F₁-score, precision and recall respectively of 73.88%, 79.24% and 69.62% (see table 4.11). All three algorithms produced relatively similar results: CRF performed slightly better than SVM (+1.95%) and moderately better than MaxEnt (+3.45%). The breakdown of the evaluation results by entity type reflects a very similar situation (table 4.12). CRF outperforms the other models with regards to the extraction of two entities out of four (i.e. Awork and Refauwork), while SVM yields the best results for the two remaining entity types (i.e. Aauthor and Refscope).

Table 4.11: Evaluation results for the named entity extraction: overall precision, recall and F₁ score of CRF, MaxEnt and SVM.

Algorithm	Precision	Recall	F ₁ Score
CRF	79.24%	69.62%	73.88%
MaxEnt	75.29 %	66.75%	70.43%
SVM	74.44%	70.21%	71.93%

Remarkably, the accuracy with which the names of ancient authors are extracted (i.e. F₁ score of Aauthor entities) is much lower than the rest of the entities across all three models, with a resulting negative effect on the overall accuracy. More precisely, all models fail to capture a high number of author names, as the lower values of recall indicate. Looking more closely at the errors, it is possible to observe that author names tend to be missed more often when they do not occur in proximity to a work title or a canonical reference, and this also happens when the author name is exactly the same. One possible solution to the low recall of aauthor entities would be to extract some additional features that can help capture what characterises author names as opposed to words that are not proper names.

³⁸ CRF++, <http://taku910.github.io/crfpp/>.

³⁹ Scikit-learn, <http://scikit-learn.org/>.

⁴⁰ I have used version 0.15.2 of the scikit-learn library and version 0.55 of CRF++.

Table 4.12: Evaluation results for the named entity extraction: precision, recall and F_1 score of CRF, MaxEnt and SVM, divided by named entity type.

Entity	CRF			SVM			MaxEnt		
	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1
Aauthor	91.15%	39.67%	54.53%	86.51%	43.61%	57.79%	90.07%	36.96%	52.13%
Awork	96.54%	71.04%	81.60%	96.00%	69.33%	80.23%	95.71%	66.55%	78.14%
Refauwork	91.72%	76.56%	83.19%	86.70%	71.53%	78.04%	84.80%	64.81%	73.16%
Refscope	96.24%	77.65%	85.48%	94.59%	80.20%	86.60%	95.55%	78.23%	85.83%

4.5.2 Evaluation of Relation Detection

As illustrated in section 4.3.1, citations are represented in the annotated datasets as *binary relations* between named entities. Such relations are binary meaning that exactly two entities are involved in any relation; each of the entities involved is called an *argument* of that relation. In the annotation environment that was used, Brat, each relation is represented visually as an arrow connecting its two arguments (see figure 4.14).

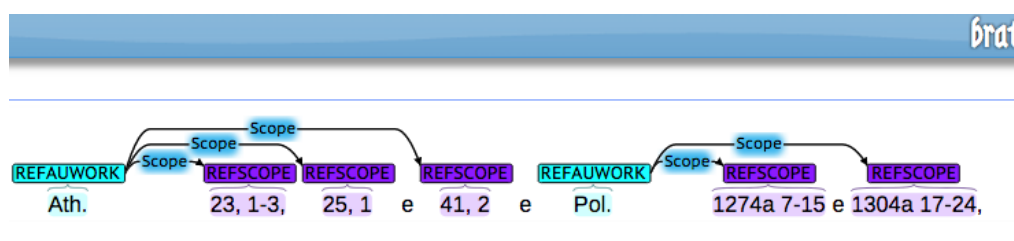


Figure 4.14: Example of scope relations represented as binary relations between named entities and visualised in Brat.

The object of the evaluation presented in this section was the rule-based algorithm for the automatic detection of scope relations that was described above (see algorithm 1). The accuracy of this algorithm was evaluated based on the 381 relations contained in the 366 documents of the manually corrected dataset. The achieved precision, recall and F_1 score are respectively 93.33%, 91.87% and 92.60% (table 4.13). Given the fairly simple set of rules used by the algorithm – which currently uses as the sole criterion to detect relations the order in which entities appear in the text – the automatic detection of relations did not prove to be a particularly problematic processing step. The algorithm failed to detect approximately 8% of the total relations as indicated by the number of false negatives. Similarly, 6.5% of the extracted relations are erroneous

meaning that they were captured by the algorithm although they were not contained in the manually corrected data (i.e. false positives).

Table 4.13: Evaluation results for the relation detection: precision, recall and F_1 score of the rule-based algorithm.

True Pos	False Pos	False Neg	Precision	Recall	F_1 Score
350	25	31	93.33%	91.87%	92.60%

Looking at the actual mistakes made by the algorithm, and specifically at the missed relations, it is possible to observe that, as had already been anticipated, they are a relatively small number of cases where the relation proceeds from right to left instead of the relatively more common left-to-right order. The direction of the relation is determined by the order in which its arguments appear: left-to-right when the Refscope entity comes first and right-to-left when an entity of a different type comes first as happens in the vast majority of cases. A left-to-right scope relation characterises citations that are expressed in a discursive way: the reference “les v. 9–12 des « Acharniens »”, for example, can also be expressed in a more concise and less discursive way as “Ar. *Ach.* 9–12”. Some examples of the kind of relations that were missed by the algorithm are provided in table 4.14.

Table 4.14: Some examples of the relations that result from discursive citations.

Example	Relation	Example with POS Tags
du [REFSCOPE chant 4] de l’ [AWORK « Énéide »]	scope(“« Énéide »”, “chant 4”)	[PRP:det du]/[NOM chant]/[NUM 4]/[PRP de]/[DET:ART l’]/[PUN:cit «]/[NOM Énéide]/[PUN:cit »]
Le [REFSCOPE livre 13] de la [AWORK « Chronique »]	scope(“« Chronique »”, “livre 13”)	[DET:ART le]/[NOM livre]/[NUM 13]/[PRP de]/[DET:ART la]/[PUN:cit «]/[ADJ Chronique]/[PUN:cit »]
les [REFSCOPE v. 9–12] des [AWORK « Acharniens »]	scope(“« Acharniens »”, “v. 9–12”)	[DET:ART les]/[ABR v.]/[NUM 9–12]/[PRP:det des]/[PUN:cit «]/[NOM Acharniens]/[PUN:cit »]

One way of capturing such relations would be to add to the algorithm a rule that parses entities in a right-to-left order. Another solution, perhaps more robust, would be to employ a supervised learning approach and harness the common pattern of PoS tags that characterises these

relations. The sequence of PoS tags could be one of the features that are used to train a statistical model to detect relations from text.⁴¹

4.5.3 *Evaluation of Entity and Relation Disambiguation*

Similar to the evaluation presented in the previous section, evaluating the disambiguation of entities and relations did not require a cross-evaluation strategy. In fact, the solution being evaluated did not involve any machine learning algorithm (section 4.4.4).

What was evaluated is the disambiguation of names of ancient authors (i.e. Aauthor entities), titles of their works (i.e. Awork entities) and references to specific sections of these works (i.e. scope relations). Approximately 55% of the 855 disambiguations contained in the training set concern authors and works, while the remaining 45% are disambiguations of canonical references. As was described above, the disambiguation of such references as well as of the mentions of authors and works is done by assigning the corresponding CTS URN to the entity that is being referred to.

Before considering the results in detail, it is worth highlighting that disambiguation is the processing step in which having an exhaustive knowledge base matters the most. Since an essential part of the disambiguation process consists of trying to match strings extracted from the text against lists of names, titles and abbreviations, the more complete such lists are the more accurate the matching is going to be. However, as I shall clarify later in the section, a more accurate matching does not necessarily guarantee better overall performances. In fact, to disambiguate correctly references that are ambiguous (i.e. they may be matched exactly but they refer to several entities) string matching needs to be complemented with ways of modelling the context where the string occurs.

Both exact and approximate matching to disambiguate entities and relations were evaluated as summarised in table 4.15.⁴² Two different threshold values were employed to measure the similarity between strings by means of edit distance (i.e. $n = 4$ and $n = 7$). Using a threshold value of 4, for example, means that matches with an edit distance

⁴¹ For an overview of features that can be used to perform relation detection in a supervised machine learning setting see Jurafsky and Martin (2009, pp. 768–772).

⁴² For an explanation of the difference between exact and approximate matching see section 4.4.4.

greater than 4 are discarded. Increasing the threshold is particularly useful to match those names or titles that are not yet contained in the knowledge base and, in general, proves to be more useful for longer strings, such as work titles consisting of several words. The effects of using a different threshold are reflected in the results and specifically in the recall.

Table 4.15: Evaluation results for the disambiguation of aauthor and awork entities and scope relations.

Matching Type	Precision	Recall	F1-Score
Exact	58.33%	62.88%	60.52%
Approximate (threshold=4)	61.04%	90.94%	73.05%
Approximate (threshold=7)	58.94%	94.76%	72.67%

The best performances were achieved by using approximate string matching with a threshold of 4: this led to an improvement of +12.53% over the exact matching and was not outperformed by using a higher threshold. The highest recall (94.76%) was achieved with a threshold value of 7, meaning the system can disambiguate some references that would otherwise have been missed. However, this higher recall came at the price of a lower precision (58.94%): in some cases it is correct that no match is found in the knowledge base. With a higher threshold the system also tended to find a match for those entities that should not have one. For instance, the reference « Lettera ai Romani » (i.e. *Epistle to the Romans*) is correctly interpreted by the system as being a reference to an ancient work (i.e. Awork entity) but this work is not contained in the knowledge base as it falls outside the scope of classical works. However, using approximate string matching with a threshold of 7 leads to additional but unrelated matches like “lettera ad erodoto”, “lettere di contadini” and “lettera di barnaba”.

What is remarkable about the results shown in table 4.15 is the relatively low precision of the system in comparison with the recall. In roughly 40% of the cases the system makes an incorrect guess concerning how a given entity or relation should be disambiguated, as the number of true positives and false positives shows. In order to identify the reason for such a low precision, it is useful to distinguish between the disambiguation of entities and the disambiguation of relations. As can be seen in table 4.16, excluding the relations from the evaluation leads to better results and to a levelling of precision and recall. This indicates

that the system fails in determining the right disambiguation for scope relations (i.e. citations) more often than when disambiguating entities.

Table 4.16: Evaluation results for the disambiguation of aauthor and awork entities only.

Matching Type	Precision	Recall	F ₁ Score
Exact	92.28%	54.56%	68.58%
Approximate (threshold=4)	82.42%	88.44%	85.32%
Approximate (threshold=7)	79.04%	93.32%	85.58%

In the results presented so far the the disambiguation of relations has emerged a particularly problematic task. At this point, it is worth recalling how the disambiguation works in the current implementation. Consider for example the citation “Thuc. I 89, 1s.”, which is represented as a relation of type scope between two entities. The first step is to determine which work is referred to by the string “Thuc.”, which corresponds to a Refauwork entity; the second step is to normalise the notation “I 89, 1s.”, which is the scope of the reference and is therefore captured by a Refscope entity, into “1.89.1–1.90.1”.

The first type of error committed by the system specifically concerns abbreviations. Consider the following example:

But **Horace** undermines the suggestion that his own poetry will forever represent the Augustan Age. **Carm. 4, 15** in fact [...]

The system fails to determine that the abbreviation “Carm.” refers to the *Carmina* by Horace. The right choice here would probably be obvious to the reader: the name of Horace, which has been mentioned in the previous sentence, provides the *context* necessary to know that “Carm.” refers his *Carmina*. Unlike the human reader, the program – at least in its current implementation – attempts to disambiguate the abbreviation “Carm.” without relating it to the named entities contained in the previous or following sentences. As a result, a lookup of the string “Carm.” returns 75 exact matches (i.e. matches with edit distance equal to zero). As for the system all such matches are equally plausible, the first one is picked thus leading to an incorrect disambiguation.

Similar errors are committed by the system when highly ambiguous abbreviations such as “ann.” and “ep.” are encountered; these abbreviations, which stand respectively for *Annales* and *Epistulae*, similarly to

“carm.” can refer to several works and require us to consider the surrounding context in order to be correctly disambiguated (see table 4.17). One relatively straightforward way of avoiding similar errors would be to let the system take into account the preceding named entities when disambiguating a given abbreviation.

Table 4.17: The 10 most ambiguous abbreviations of ancient works in the knowledge base. The ambiguity of an abbreviation is measured based on the number of unique works it may refer to.

Abbreviation	Meaning	Unique Works
“epigr.”	Epigrammata	96
“carm.”	Carmina	75
“ep.”	Epistulae	74
“or.”	Orationes	73
“orat.”	Orationes	71
“epist.”	Epistulae	66
“hist.”	Historiae	37
“gramm.”	Grammatica	36
“ann.”	Annales	16

The second type of error committed by the system concerns ambiguous author names. The mention “Aristophanes”, for example, may refer to the comic playwright as well as to the Alexandrian grammarian Aristophanes of Byzantium. Similarly, “Pliny” may refer to Pliny the Younger or Pliny the Elder. Again, the method of ascertaining which of the possible entities is actually being referred to is to look at the surrounding context. Consider the following passage drawn from the beginning of the review of an article entitled *I grammatici alessandrini nei papiri di Aristofane*:

Esame dell’ esegesi papiracea ad **Aristofane** : permanenza del lavoro degli eruditi alessandrini [...]

Keywords such as “grammatici”, “eruditi”, “alessandrini”, “marginalia”, “hypomnemata”, “commentari”, “tardo-alessandrini”, which are contained both in the title and in the review, leave no doubt to the reader about the fact that the Aristophanes referred to in this context is Aristophanes of Byzantium.

Enabling the system to perform a similar operation – i.e. disambiguating a name based on words that occur in the surrounding text – would require us to make use of the *context*. In fact, the effect of the context in

natural language communication is to constrain interpretation or, in this case, to allow us to understand a reference despite its apparent ambiguity. However, as Hirst (1997) convincingly argued, creating a formal model of context is impossible as context is always constructed by the speaker (or author) and the interpreter. Therefore, given the impossibility of modelling the context, it is just possible to create something that approximates it, a surrogate.

One possible way of mimicking to some extent how context works would be to build for each entity in the knowledge base a list of co-occurring words. In order to extract words that are distinctive it is possible to remove the so-called *stopwords* first and then to retain only words that have a relatively low frequency within the training dataset. Moreover, in order to allow for multilingualism, one list of related words for each language is required; alternatively, one could maintain such a list in one language and then use language alignment to compare sequences of words written in other languages with this list.

The third type of error consists of a limited number of cases characterised by implicit topicalisation. In such cases the information necessary to disambiguate a citation is not contained in the text of the review but in the title of the reviewed publication. Since the title of the reviewed publication is by design not included in the dataset as is part of the metadata, such errors highlighted a limitation related to how the datasets have been constructed. An example of this type of error can be seen in the following passage:

Dans son **chap. 5** sur le squelette et la respiration, **Lactance** utilise des sources disparates et arrive aux limites de son savoir médical.

Without looking at the title of the article – “Lactance, *De opificio Dei* (303–304): le savoir médical au début du IV^e siècle” – it is not possible to guess to which of Lactantius’ works does the reference to chapter 5 refer.

The final error type concerns references that were expressed ambiguously. In such cases, which proved to be challenging not only for the programme but also for the human annotators, determining the correct answer is not always possible and when it is possible requires some additional knowledge about the cited works. Consider for instance the following passage:

Analysis of the pederastic poems in the Theocritean corpus (12 ; 23 ; 29 ; 30) reveals that **Theocritus** reflects on mutuality in a relationship [...]

Since two collections of poems are attributed to Theocritus – the *Idylls* and the *Epigrams* – the expression “pederastic poems” could equally refer to either of them. The human reader, however, will know – or at least is expected to know in this example – that this passage cannot refer to the *Epigrams* as only 22 epigrams are attributed to Theocritus. As this example shows, rhetorical choices with regards to how references are expressed made by the author make the task of automatically capturing such references considerably more challenging. In order to cope with such references it is necessary to build into the system some sort of reasoning capabilities that uses the data contained in the knowledge base: in the example below, the system could rule out the *Epigrams* after having checked that the references “29” and “30” are only valid in relation to the *Idylls*.

4.6 DISCUSSION AND FURTHER WORK

The results of the evaluation show that named entity extraction does provide a suitable framework to *translate* the extraction of canonical citations into a computationally tractable problem. This is what motivated my research in the first place. The evaluation results should be considered in light of the main research goal, which was to test *if, how* and *how well* the task of identifying, extracting and indexing canonical citations can be automated and not to create a programme that would achieve the best possible results in terms of accuracy in performing such task. As observed above, a limitation of the evaluation is that it was not possible to compare the performance of the system with a baseline due to the lack of comparable systems.

The importance and value for classicists of having a system to extract citations from text automatically has already been explained. One might ask, however, what the relevance is for a broader audience and more generally what contribution is being made to the field of NLP. I argue that performing citation extraction as a pre-processing step allows for greater accuracy in carrying out virtually any NLP task on texts coming from the domain of Classics. Indeed, abbreviations, which are exten-

sively used within canonical references, are often responsible for a high number of errors in the very first steps of the text processing pipeline, such as sentence segmentation or tokenisation. Such errors, if they are not corrected, are then carried forward to the subsequent – and often more complex – processing steps such as syntactic parsing. Consider the following sentence drawn from a scholarly article in German:

In seiner Beschreibung der politischen Verfassung Athens zur Zeit Solons bezeichnet Aristoteles (Ath. Pol. 7, 4) die Akropolisvotive jener Zeit als ἀναθήματα τῶν ἀρχαίων (»Weihgeschenke der Ahnen«) und verwendet ein archaisches Weihgeschenk explizit als historisches Zeugnis.

As shown in figure 4.15, the fact that the reference to Aristoteles' *Athenaion Politeia* ("Ath. Pol. 7, 4") is not correctly captured – and, more generally, the fact that "Ath." and "Pol." are not recognised as being abbreviations – leads to the incorrect tokenisation of the abbreviations contained in the sentence. This error, in turn, leads to the sentence boundary being placed in the middle of the citation with the resulting truncation of the syntactic tree of the overall sentence.

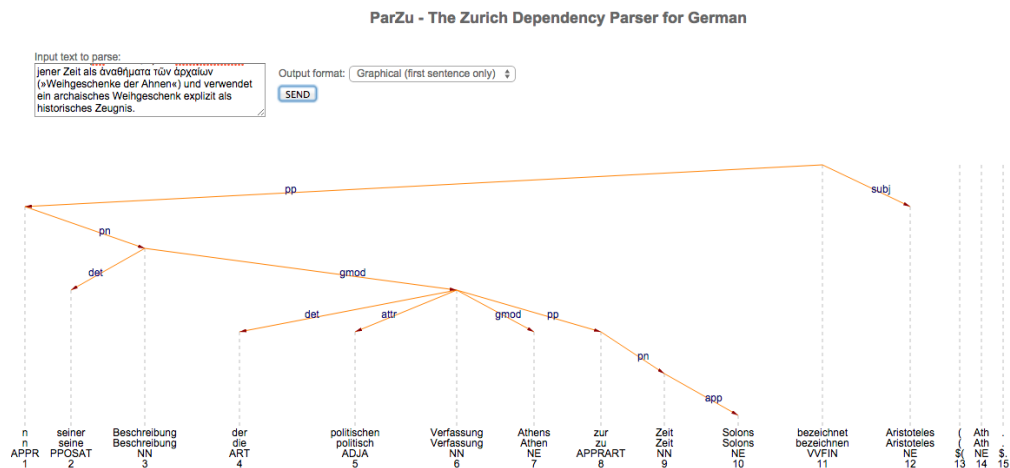


Figure 4.15: The syntactic tree of a sentence containing an incorrectly tokenised abbreviation.

What the evaluation results showed is that the programme developed as part of my research correctly extracts canonical references provided that: a) the reference is expressed in a structured rather than discursive way (e.g. "Hom. *Il.* 1, 1–10" as opposed to "in the prolog of the first book of the *Iliad* (vv. 1–10), Homer"; b) a consistent use of punctuation

signs is made within the reference and c) sufficient information concerning the cited text is contained in the knowledge base. That being said, substantial room for improvement exists to enhance the accuracy with which the extraction of canonical citations can be performed as I shall discuss next.

The first area of improvement is the population of the knowledge base with data about how to refer to classical texts as several processing tasks – i.e. sentence segmentation, text tokenisation, named entity recognition as well as disambiguation – can benefit, to a varying extent, from a more comprehensive knowledge base.

The second area of improvement concerns the application of machine learning algorithms to each of the three steps of citation extraction. In the current implementation of the system, the only step to which a machine learning-based approach has been applied is the recognition and classification of named entities. However, applying machine learning to the two remaining steps is not only possible but also desirable and interesting in itself insofar as it would allow us to compare such an approach with the rule-based approach that I have adopted.

Moreover, the accuracy of the named entity extraction can be improved in at least two respects. First, the number of features that are observed for each token can be increased by introducing new features. In addition to this, it is possible to select those features that prove to be most effective to learn the statistical model, the so-called *feature selection*. Second, the statistical models can be fine tuned by setting the parameters of each statistical model to different values in order to find the optimal combination of parameter settings.

The third (and last) area of improvement is related to expanding the training set, a task that overlaps substantially with continuing to grow the knowledge base. As new data is manually corrected and included into the training set, new pieces of information are found that can be fed back into the knowledge base by means of largely automated processes. Training data, in turn, could be expanded with regards to both its breadth and depth.

Expanding the breadth of the training set means to increase its size while trying to cover as many styles of citing primary sources as possible. The use of angle quotes to cite titles of ancient works (e.g. “Homer’s « Iliad »”), which is consistent throughout the APh, is a good example of a convention that is not common in every journal or publication in

Classics. Therefore, in order for the training set to be generalisable – i.e. suitable to support the extraction of canonical citations no matter the citation style they follow – the range of citation styles that are represented in the corpus needs to be as comprehensive as possible.

Expanding the depth of the training set involves extending the annotation scheme. As pointed out above, the accuracy with which the named entity classification is performed could be improved by introducing new types of named entities. These new entities would be especially useful to distinguish those strings that are most often confused with canonical references due to their surface similarity (e.g. references to papyri, citations of fragmentary texts, etc.). In addition to introducing new entities, the depth could be extended by adding new layers of linguistic annotations such as chunking and syntactic annotation. These additional layers of annotation can then be harnessed by using them as features while training a statistical model to learn the classification of named entities or the detection of relation between entities.

Moreover, adding the layer of syntactic annotations would allow us to capture the indication of the specific phenomenon about which a given text passage is cited. Consider the following sentence:

Ammianus' conception of how to deal with foreigners is based on Vergil's formulation of Rome's mission (Aen. 6, 851-853).

The syntactic parsing of this sentence allows us to identify the noun phrase "Vergil's formulation of Rome's mission" and to establish that the canonical reference given in brackets "(Aen. 6, 851-853)" actually refers to it. Capturing this kind of information is necessary as a first step towards an automatic classification – or at the very least characterisation – of canonical citations. Being able to classify such citations based on what they are cited for becomes more and more important as the number of references that are extracted grows.

4.7 SUMMARY

In NLP the task of capturing the named entities mentioned in a text is called NER. It consists of three steps: 1) the extraction of the named entities; 2) the detection of the relations between them and 3) the disambiguation of the extracted entities and relations. The approach I have

developed to the extraction of canonical references aligns with such a three-step process. I have defined four named entities that capture the various citation components. I have then defined a canonical citation as a binary relation between two named entities. Thus, the extraction of canonical references consists of the following steps: 1) identifying the citation components; 2) combining these components into references and 3) disambiguating these references by means of CTS URNs. A system implementing this approach was developed and evaluated against a manually corrected sample of APh abstracts. This system uses a machine learning-based approach for the first step, while the two remaining steps are implemented using a rule-based approach. The accuracy (i.e. F_1 score) of this system for each of these processing steps is: 73.88% (named entity extraction); 92.60% (relation extraction) and 73.05% (entity and relation disambiguation).

CITATION NETWORKS AND THE STUDY OF CLASSICAL TEXTS

Overview

Once canonical references are extracted from texts, the web of relations that these references implicitly constitute can be formalised as a citation network. Such a network lends itself to be analysed, visualised and searched. In particular, this chapter focusses on how such a network can be exploited as a means of searching through secondary literature in a way comparable to what indexes of cited passages – *indices locorum* – already allow.

In section 5.1 I consider the implications of shifting from manually created indexes to automatically extracted citation networks. Section 5.2 introduces the key concepts of citation networks and provides an overview of network approaches to the analysis of citations. Finally, in section 5.3 I discuss how these citation networks are constructed and how they can be used to find information within secondary literature.

5.1 FROM INDEX LOCORUM TO CITATION NETWORK

The automatic extraction of canonical references presented in the previous chapters can be seen as a means to automate, with some degree of accuracy, the creation of indexes of cited passages – *indices locorum*. In fact, having the canonical references available as a semantic graph – a set of instances of ontology classes connected by their properties – opens up new ways of making use of these references. In particular, this chapter illustrates how the web of relations between publications and primary sources can be formalised as a citation network, i.e. a formal model of the relations that exist between the nodes constituting the network.

Shifting from manually created indexes of cited passages to automatically extracted citation networks does have significant implications. These primarily emerge from the scale and accuracy of this new type of data as well as the potential for new insights into texts.

5.1.1 *Scale and Accuracy*

The first difference between *indices locorum* as we currently know them and these citation networks concerns the scale of materials that can be indexed and the overall accuracy of the resulting index.

Having an index of cited passages is currently regarded as a highly valuable feature of a scholarly publication. This is due both to the usefulness of a tool which allows the reader to find cited passages in a publication and to the high cost of creating it. Despite the convenience of using modern word processors to partly automate this process, creating an index relies heavily on manually marking the portions of text to be indexed. As a result, such indexes tend to be highly accurate, however they are inevitably selective as they can cover just a fraction of what is published in Classics.

In contrast, the automatic extraction of canonical references makes it considerably faster to index publications, including those such as journal articles that would never have been given consideration otherwise.¹ As a result of this increase in speed, it becomes possible to index cited passages on a much larger scale. The availability of automatic indexing systems and of large scale digital archives means that we are no longer forced to consider just a *sample* of publications but can now systematically index *entire* archives. The JSTOR archive is a good example of this. Even though it is not entirely comprehensive, being able to index the hundreds of thousands of journal articles contained within it constitutes a remarkable change in scale.

Such a change, however, comes at the price of a loss of accuracy. In fact, as shown by the evaluation of the citation extraction system discussed in section 4.5, the extraction and disambiguation of canonical references is far from being as accurate as current *indices locorum*, al-

¹ It takes approximately 24 hours (of computing time) to index a volume of the *L'Année Philologique* (APh) on a laptop machine and about the same time to index the full text of 1,000 journal articles from JSTOR. However, these figures can be significantly reduced by running the indexing in parallel using grid or cloud computing facilities and by optimising the code.

though as I mentioned, there is room for substantial improvements. Nevertheless, the ability to measure the accuracy of automatic indexing means that overall accuracy can be explicitly taken into account when analysing and interpreting the results.

5.1.2 *Manipulability*

A second substantial difference is the manipulability of the underlying data that characterises a citation network as opposed to a traditional *index locorum*. The latter is static representation of a list of locations within a publication where a given text passage is cited. In contrast, a digital index in the form of a citation network not only allows us to create multiple representations and visualisations of this data, but it also allows us to manipulate it in several ways.

The manipulability of the underlying data that characterises digital indexes as opposed to print indexes enables the quantitative exploration of the citation data they capture. For example, it becomes possible to compute the frequency with which a given text passage is cited. If the citation data at hand covers a wider temporal span it is possible to observe how this frequency varies over time, thus providing some insights into the diachronic variation of the number of publications that have discussed a given text passage. Moreover, since for each extracted citation additional information about its author and work is retained, citation counts can be aggregated at the level of author or work.

5.1.3 *Networks of Relations*

A third difference concerns the nature of the view on texts that characterises an index as opposed to a citation network. On the one hand, an index and a network are two equivalent ways of representing the same information. This is proved by the fact that, in some cases, networks were derived and constructed out of digitised indexes.² On the other hand, however, an *index locorum* presents to the reader the cited passages in isolation, whereas a (citation) network emphasises by its

² An example of this approach is given by Rochat (2014) who has recently used the index of characters from an edition of Roussaeau's *Les Confessions* to study the character network in this text.

very nature the relations that exist between publications and between the cited primary sources.

This isolation, which is overcome in a citation network, is two-fold. First, each index provides access to a specific publication and is not related to similar indexes of other publications. Second, each cited passage corresponds to a separate entry in the index without being related to other passages cited in the same context. In contrast, publications citing the same author, work or text passage are connected together and shown in relation to each other in a citation network. In this respect, a citation network may be seen as separate indexes of citations, one per publication, stitched together to form a network of relations.³

Given this emphasis on relation over isolation, a citation network is especially suitable for capturing the aspect of intertextuality that is reflected in the web of relations that canonical references constitute.⁴ A central tenet in intertextuality is that literary texts are to be read in light of their relation to and position within the literary system that they form.⁵ It is worth pointing out, however, that such a citation network does not capture exclusively nor directly the intertextual relations between texts. Instead, it captures and represents the traces of these relations that scholars leave in their publications in the form of canonical references.

5.2 NETWORK APPROACHES TO CITATIONS

This section aims to put citation networks into the broader context of network theory, to introduce the basic concepts of citation networks and to discuss related research in this field.⁶

³ At the technical level, this “stitching indexes together” is enabled by the use of machine-readable identifiers as opposed to human-readable strings to express references to authors, works and text passages. In the case of this research the identifiers used are the Uniform Resource Names (URNs) defined by the Canonical Text Services (CTS) protocol (see *infra* at p. 78).

⁴ More generally, the adoption of a network approach in the field of Digital Humanities (DH) has become increasingly common over the last few years. An indicator of such a trend is the increased rate of network-related papers submitted to the annual DH conference over the last 3 years (Weingart, 2014).

⁵ The concept of a *literary system* was introduced to Classics by Conte (1974) and was further elaborated by Fowler (1997).

⁶ For an introduction to networks see Newman (2010) and Easley and Kleinberg (2010). Newman (2010) discusses citation networks in chapter 4 while Easley and Kleinberg (2010) consider different aspects of these networks in chapters 2, 13 and 14.

Citation Networks and Network Science

Citation networks belong to the broader category of *information networks*, namely “networks consisting of items of data linked together in some way” (Newman, 2010, p. 63). The most well-known of this category is probably the network of documents that constitute the World Wide Web.⁷

The main assumption for the study of citation networks is that two documents are *somehow* related if they cite each other or, in other words, that citation networks represent “networks of relatedness of subject matter” (Newman, 2010, p. 68). Although citations between academic publications are the most studied type of citations, other types include legal networks consisting of citations between such things as legal cases or between patents.

The Basics of Citation Networks

In a network entities of interest are represented as nodes and edges connecting these nodes.⁸ In a citation network of modern publications, each node represents a publication while an edge between any two nodes indicates a citation between the corresponding publications (figure 5.1 a).

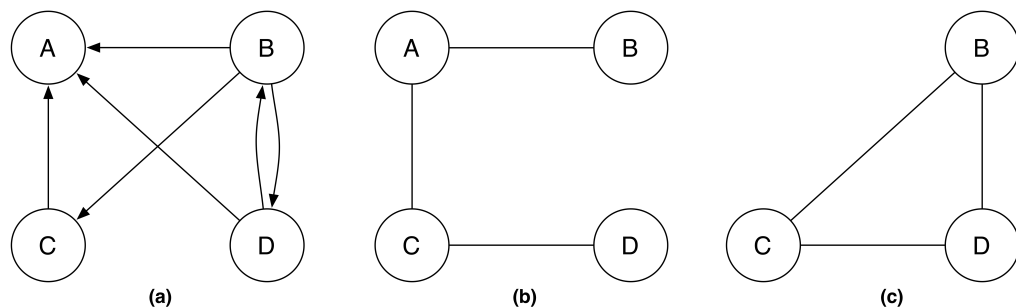


Figure 5.1: The figure shows a directed citation network consisting of four publications (a) and two undirected co-citation networks derived from it: co-citation (b) and citation coupling (c).

The relation between the two publications that a citation constitutes is asymmetric: the fact that B cites A does not also imply that A cites B. For this reason, citation networks are *directed*, meaning that the direc-

⁷ In addition to information networks, other categories of networks being investigated in the field of network theory are 1) *technological networks* such as power grids or transport networks; 2) *social networks* such as those emerging from email communications or affiliation networks; 3) *biological networks* such as networks of interactions between proteins or neural networks.

⁸ Nodes are also called *vertices* and edges are sometimes also referred to as links.

tionality of the edges matters. There are also symmetric relations such as friendship that are represented as *undirected networks*.

In addition to directionality, edges can have a *weight*, i.e. a numerical value qualifying some aspect of the relation that the edge represents. In the context of a citation network, for example, the edge weight can indicate the number of times publication B cites publication A.

Two other types of (undirected) networks can be projected – i.e. derived – from citation networks. First, a *co-citation* network where two publications are related if they are both cited by a third publication. Second, a *citation coupling* network where two publications are related if they both cite a third publication. An example of these two networks is shown in figure 5.1 b-c. It is interesting to note that the citation network which can be constructed from canonical references, as discussed later, is very similar to a citation coupling network, the only difference being that two publications are related when they cite the same primary source rather than the same secondary source.

Citation Network Studies

Research on academic citation networks began in the 1960s in the field of bibliometrics, the branch of Library and Information Science (LIS) that uses statistics to investigate the publication process and to evaluate research achievements.⁹ One of the most important contribution that bibliometrics has made is the manual creation of bibliographic databases and citation indexes such as the Science Citation Index (SCI) and the Arts and Humanities Citation Index (A&HCI).¹⁰ These datasets have made it possible to study citation networks.

More recently, digitisation initiatives, automatic citation indexing and increased computational power have made possible the collection and analysis of large amounts of citation data. The availability of large amounts of citation data, in turn, has laid the foundations for the study of large-scale citation networks in the field of network science.¹¹

The assumption for the analysis and study of these networks is that they constitute a proxy for the understanding of knowledge dynamics such as the spread of ideas across disciplines. Brughmans (2013), for example, has recently employed citation network analysis to study how ar-

⁹ See (Franceschet, 2012, 838–839) for a historical perspective on the study of citations in bibliometrics and network theory.

¹⁰ On these indexes see *infra* at p. 19.

¹¹ See Radicchi et al. (2012) for an insightful overview of citation network studies.

chaeological research adopted and used formal network methods from other disciplines.

Moreover, an important area of investigation is concerned with the internal organisation of networks into communities (or clusters). At a high level of abstraction, the analysis of these communities can provide some insight into the flow of information between disciplines.

Moreover, other kinds of networks can be constructed starting from citation data. An example is the collaboration network between scholars that can be derived from the authorship information (i.e. two scholars are connected if they have published together). Franceschet (2011), for example, used this kind of social network to show how scholars in computer science have become increasingly connected over the last 50 years.

While the study of citation networks has been successfully applied to the sciences, what has proved more problematic has been its application to the humanities. This has been due to the lack of citation data in the humanities as compared to the sciences but also, more importantly, to the different contexts in which citations are deployed within humanities discourse (Sula and Miller, 2014). These contexts tend to be neutral in the sciences as citations are mainly used to refer to past work, whereas in the humanities they are more often characterised either in a positive or negative way. In fact, scholars in the humanities may cite others' work to express agreement or praise but also to criticise or refute it.

Given the density of citations to primary sources that publications in the humanities contain, it is surprising that the study of networks consisting of citations between modern publications and primary sources remains a largely unexplored area. One example from this area is the work by Murai and Tokosumi (2005) and Murai et al. (2008). They have focussed in particular on canonical references to the Bible that are found within theological writings. Their analysis of the co-citation network of these references – i.e. which text passages of the Bible are cited in relation to each other – highlighted different conceptualizations of Christian dogma.

5.3 TEXTS THROUGH THE LENS OF A NETWORK

This section describes the decisions I have taken to represent the web of canonical references as a formal network. I then consider how this

network can be used to search through publications by taking as an example a volume of APh. Finally, I discuss the advantages and limitations of accessing publications through the network of the references they contain.

5.3.1 *A Three-Level Citation Network*

Rationale

I had two goals in the creation of this citation network: first, to enable various kinds of search and, second, to enable the further analysis of the structure and properties of these networks.

The main challenge in the creation of these networks was to preserve the hierarchy of levels that is embodied in a canonical citation. Indeed, as stated repeatedly in the previous chapters, a citation refers to a specific text passage but also, albeit implicitly, to the work containing that passage and to the author of that work (e.g. Verg. *Aen.* 1.1 refers to this specific line as well as, indirectly, to the *Aeneid* and to Vergil). This specific aspect is not present in the citation networks that have been previously studied but must be addressed as it is crucial both for searching and analysing these networks.

The approach I have taken to tackle this issue, inspired by a similar approach developed by Schich and Coscia (2011), is to create a three-level network that allows us to look at the same citation data at different levels of abstraction, namely macro-, meso- and micro-level. Similar to how a lens works, these networks make it possible to produce a number of views on this data with an increasing degree of granularity and specificity. As I explain next, these different lenses are obtained by only taking into account certain hierarchical levels of a canonical reference at each level of analysis.

In particular, when creating a network careful consideration must be given to the definition of the number of node types – also called the *modes* of a network. Indeed, the number of node types affects the meaning of the network and constrains the methods and algorithms that can be used to analyse its properties and structure.¹²

¹² As Weingart (2011) rightly points out, this aspect can easily be overlooked in DH research where network methods and especially visualisation tools are adopted without the necessary understanding of the underlying theory.

Finally, the types of citations that are expressed in these networks deserve some clarification. I focussed exclusively on citations between documents (e.g. APh abstracts or journal articles) and ancient texts. Nevertheless, articles and abstracts may refer to each other as do – albeit more rarely – ancient texts. These other reference types would constitute additional networks of relations that were not captured nor considered in this study. However, studying the interplay between these two kinds of networks constitutes an area for further research. In fact, such a study could investigate to what extent articles that are citing the same subset of primary sources are also referring to each other.

Network Construction

As an example to illustrate the construction of this three-level network I shall consider the data extracted from the entire volume 75 of APh (see table 5.1). This data is synchronic as it consists of reviews of publications that appeared in 2004. Moreover, the data includes a subset of 366 documents that were processed and manually corrected in order to create a training set as discussed in section 4.3.2. The canonical references from the remaining 6,947 documents were extracted automatically and did not undergo manual correction.

Table 5.1: The table provides some statistics concerning the APh data used to construct the three-level network. The two datasets correspond to the subset of abstracts that were manually corrected (APh-vol75-gold) and the remaining abstracts that were automatically processed (APh-vol75-dev). The documents are further divided into documents where no citations were found (column ‘no-cit’) and documents where the automatic processing failed (‘err’). The number of extracted references is broken down into references to ancient authors (‘au’), ancient works (‘wo’) and text passages (‘pas’).

Dataset	tokens	Documents			References		
		no-cit	err	tot	au	wo	pas
APh-vol75-gold	25,104	131	0	366	347	171	357
APh-vol75-dev	354,672	5,342	254	6,947	1,404	800	901

References reported in table 5.1 above were referred to by different names in the previous chapters depending on the context. For example, a reference to an author corresponds to an instance of the ontology class `fbrroo:F1_Work` described in section 3.5.2 and to the entity type `Aauthor` in the annotation scheme described in section 4.3.1. These references,

represented in the data by CTS URNs, are the starting point for the creation of the citation networks.

Network Visualisations

The network visualisations that follow deserve some explanation. Although I have used only graph-based visualisations to represent the structure of the network, other visual representations of the same data are possible. For example, a graph can also be visualised as an adjacency matrix.

The layout of these visualisations is determined by the force-layout algorithm used to position the nodes on the canvas.¹³ As its name suggests, this algorithm works by applying different forces to each node in the network. These forces are: repulsion, gravity and attraction. All nodes push each other away (repulsion), whilst connected nodes are pulled toward each other (attraction). Simultaneously, gravity pushes all nodes towards the center so as to oppose the repulsion and prevent the nodes from being pushed out of sight.

This algorithm is dynamic as it runs through several iterations until a situation of balance is reached. The final configuration of the nodes results from the interplay of these three forces. As a result, nodes that are highly connected with each other tend to remain in the middle of the canvas, whereas less connected nodes are pushed towards the periphery.

These visualisations can be useful to draw attention to some high-level properties of the network. For example, since the size of a node reflects its indegree – i.e. the number of incoming connections – the most cited authors or works within the network can easily be identified. In contrast, the graph visualisation of a very large dataset may become hard to read and interpret. However, investigating what is the most suitable representation for the citation network discussed in this chapter remains a very interesting area for further research.

Macro-level Network

The macro-level network offers the most abstract view on the data and is created by treating each canonical reference as a reference to the cited

¹³ The visualisations contained in this chapter were created using the Javascript library D3.js, <http://d3js.org/> and use the implementation of the force-directed layout algorithm provided by this library.

author while leaving aside the more detailed information about which work and specific text passage are cited. For example, the references “Pliny, *nat.* 11, 4, 11” and “Vergil, *georg.* 4, 149–218” contained in the document APh 75–00113 are treated as references to the cited authors – Pliny and Vergil – when constructing this network.

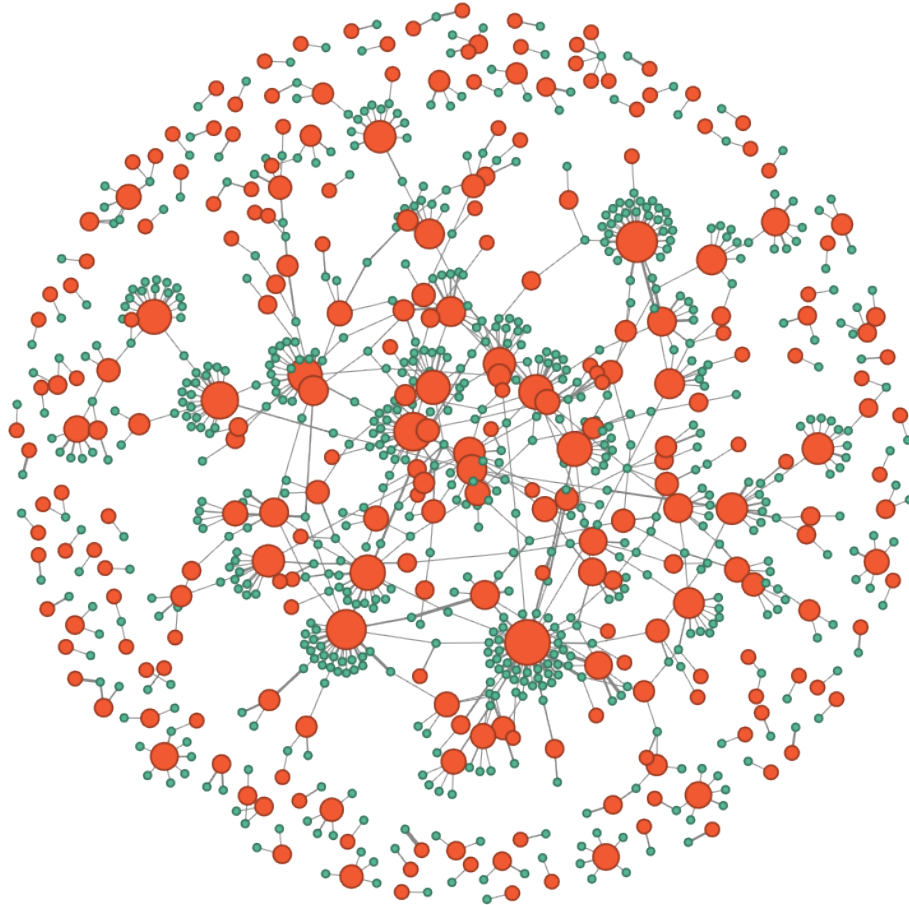


Figure 5.2: A visualisation of the macro-level citation network extracted from the APh data. The green nodes represent APh abstracts whilst the red nodes represent ancient authors. Although the directionality of the edges is not shown, the network is directed. The size of the nodes is proportional to their indegree (i.e. number of incoming citations).

Figure 5.2 shows the macro-level network for the considered volume of the APh. The network is a 2-mode (or bipartite) network since it contains two types of nodes – documents and ancient authors – and there are no edges between nodes belonging to the same type.¹⁴

¹⁴ It should be noted that while also the meso- and micro-level network are bipartite, the types of the nodes they contain do vary.

In the visualisation of figure 5.2 the size of the nodes is proportional to the number of incoming edges or *indegree*, while the thickness of the edges is proportional to their weight, in this case the number of times a given author is referred to.

The edges in this network are directed because, as was observed above, a citation can be seen as a relation going *from* the citing document *to* the cited entity – be it an author, work or text passage. Moreover, an edge in this network can have two meanings. It can mean that a given author is explicitly cited but it can also mean that the author is simply mentioned in the text. Although it is desirable to capture both cases, it is also important for the meaning of the resulting network to be able to distinguish them. When the mentions of authors are excluded, the sparseness of the network increases, meaning that the network is characterised by a smaller number of nodes and connections between them.¹⁵

Meso-level Network

The meso-level network shown in figure 5.3 offers a more detailed view of the data while maintaining some degree of abstraction compared to the micro-level network. Canonical references are not treated as references to the cited author but to the cited work. For instance, while constructing this network, the references “Pliny, *nat.* 11, 4, 11” and “Vergil, *georg.* 4, 149–218” of the example above are “compressed” respectively into a reference to Pliny’s *Naturalis Historia* and to Vergil’s *Georgics*. The meso-level network shares the same properties as the macro-level network. Indeed, it is bipartite as it consists of two types of nodes – documents and ancient works. Moreover, the edges are directed and, similar to the macro-level network, they can represent both mentions of titles of works and explicit references to specific sections of the work.

Micro-level Network

The highest degree of specificity – and thus of sparseness – is reached in the micro-level network figure 5.4. In this network each cited text passage is represented by a distinct node. The connections between nodes, as they are sparser, also become more interesting, at least from

¹⁵ The decision as to whether mentions of authors should be taken into account depends on the purpose of the network. When implementing a search application the decision may be left to the user by providing a filter that can be toggled on and off.

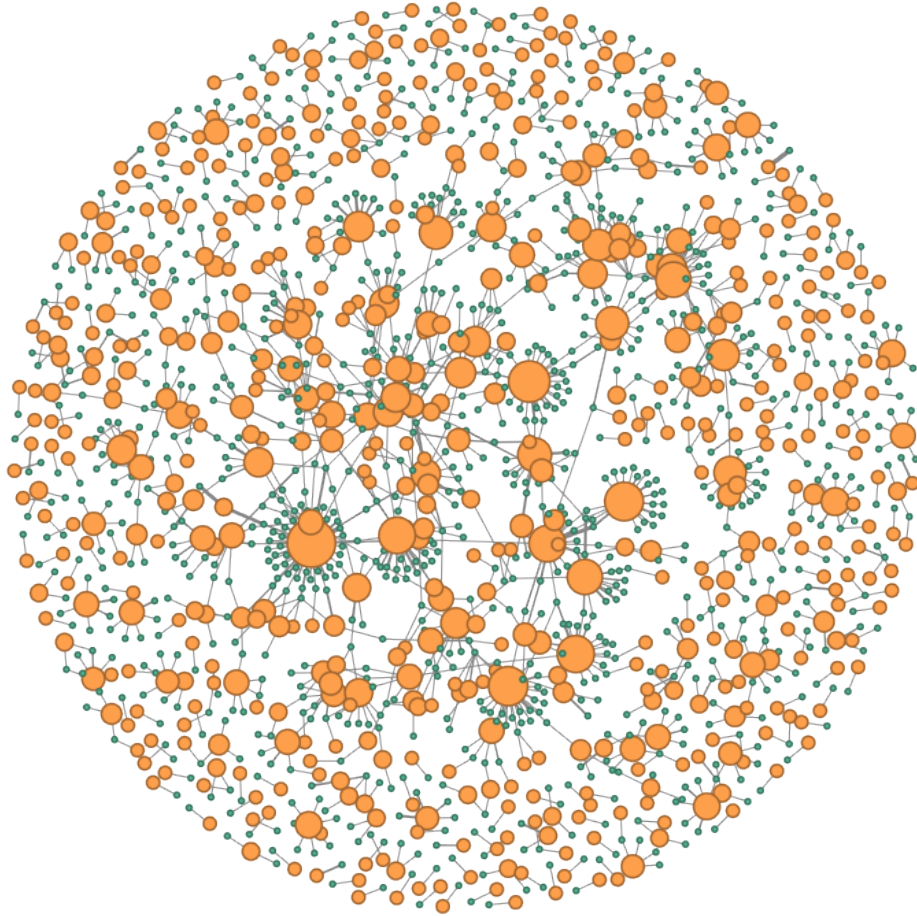


Figure 5.3: A visualisation of the meso-level citation network extracted from the APh data. The green nodes represent APh abstracts whilst the orange nodes represent ancient works.

an intertextual point of view. Indeed, while a document may refer to several text passages, the documents that refer to the very same set of text passages are most interesting as they are more likely to be closely related to each other.

5.3.2 *Network-based Search*

Let us now see how this three-level citation network can provide a means to search through a body of secondary literature. Indeed, the main motivation for this work has been that, when searching for bibliography, canonical references can serve as a valuable entry point to information.

Implementing a search application based on this citation network goes well beyond the limits of this work. However, a prototype of an inter-

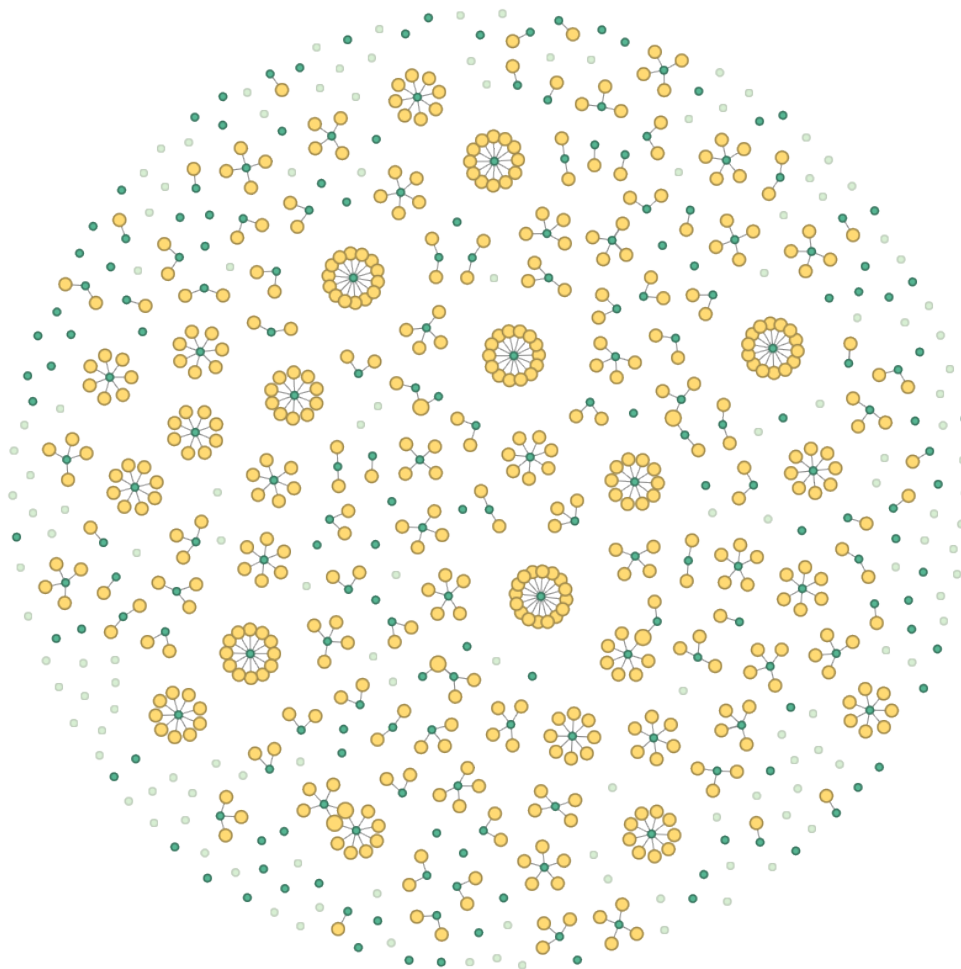


Figure 5.4: A visualisation of the micro-level citation network extracted from the APh data. The green nodes represent APh abstracts and the yellow nodes represent text passages.

active network visualisation was developed with the goal of illustrating some of the search patterns could be enabled by the properties of the underlying network. The goal of this interactive visualisation is to reproduce the dynamics of a search rather than providing a search user interface. In fact, it is possible to build a search application upon this network without even presenting the user with a graph-based search interface. Even if the search results are displayed as a list, which results are returned is determined by the structure of the network.

This interactive visualisation currently provides two functionalities.¹⁶ First, the user can select a node and see which other nodes are connected to it. If an ancient author is selected, the publications citing that author are highlighted. The nodes that are highlighted upon selection

¹⁶ An interactive version of the visualisations discussed in this chapter is available at <http://hdl.handle.net/11858/00-1780-0000-0029-9CB1-6>.

correspond to the results that a search for documents citing that author would return. Second, a double click on a node – be it a publication, an author, a work or a text passage – highlights other nodes that are connected within a certain distance. A search application could leverage this functionality by presenting the user with contents related to the initial search.

The question as to whether the graph as a visual metaphor is an intuitive way of organising a user interface for search remains and would require some user testing to be answered. An alternative approach would be to combine a graph-based visualisation with the hierarchical, tree-like structure of an index. An index and a graph, in fact, can be seen as two complementary way of accessing information. The graph facilitates browsing and allows for exploring the existing connections between resources, whereas the index provides an effective means of finding a resource already known to the user.

Search Patterns

A first set of search patterns enabled by a network-based search is similar to a search by means of an index of cited passages as it allows us to search for documents on a specific ancient author, work or text passage. However, as was noted in section 5.1, the difference is, first, the scale of materials indexed and, second, the ability to stitch, as it were, several isolated indexes together.



Figure 5.5: A screenshot of the interactive visualisation of the micro-level network extracted from the APH data. The two abstracts (green nodes) are connected as they both cite the same passage of Aristophanes' *Acharnians* (yellow node).

An example of searching for documents on a specific passage is provided in figure 5.5.

Suppose that the user is interested in the passage of Aristophanes' *Acharnians* where the tragic poet Theognis is referred to as being "cold" (Ar. *Ach.* 9–12). The search retrieves two publication reviews – out of the 6,947 contained in volume 75 – where this passage is explicitly cited.¹⁷ One publication is on the role of winter in the corpus of Aristophanes and discusses various other passages including *Ach.* 9–12. The other publication is on the reception of the comedies, and specifically on the ability of Aristophanes' audience to recognise parodic references to contemporary poets – such as in this case Theognis. In addition to retrieving the two publications citing Ar. *Ach.* 9–12, the user is presented with other Aristophanic passages that are related to it.

Moreover, what the network structure as opposed to the tree-like structure of an index does allow for is to search simultaneously for documents citing multiple authors, works and text passages. In fact, each entry of an index points to the location in the text where a given passage is cited, but does not say anything about other passages cited within the same context. For example, figure 5.6 shows the abstracts that are retrieved when *Iliad* and *Odyssey* are selected. The search returns the abstracts that refer to either work as well as those that refer to both.

A second set of search patterns is enabled entirely by the network structure. These patterns consist of extending the initial search to include further *related* elements and tend to facilitate serendipitous ways of discovering documents that are relevant to one's search.¹⁸ Such patterns are especially suitable to support the search by browsing through content rather than searching for something specific.

What is leveraged to expand the initial search are the connections between nodes in the network. For example, a search for documents citing Vergil can be expanded to include other authors cited by documents

¹⁷ It is worth observing that in the specific case of the APh the canonical references are those found in the abstract of the reviewed publication. Ultimately, it is the reviewer who selects which passages – among the many cited within the publication – should be mentioned explicitly to summarise the contribution of the publication.

¹⁸ The related nodes are extracted from the network by using the shortest path algorithm. The algorithm computes the distance between two nodes in a network as the number of 'hops' needed to go from one node to the other. In the examples discussed in this section, an all-pairs shortest path algorithm is used to find all other nodes connected to a given node within a certain distance.

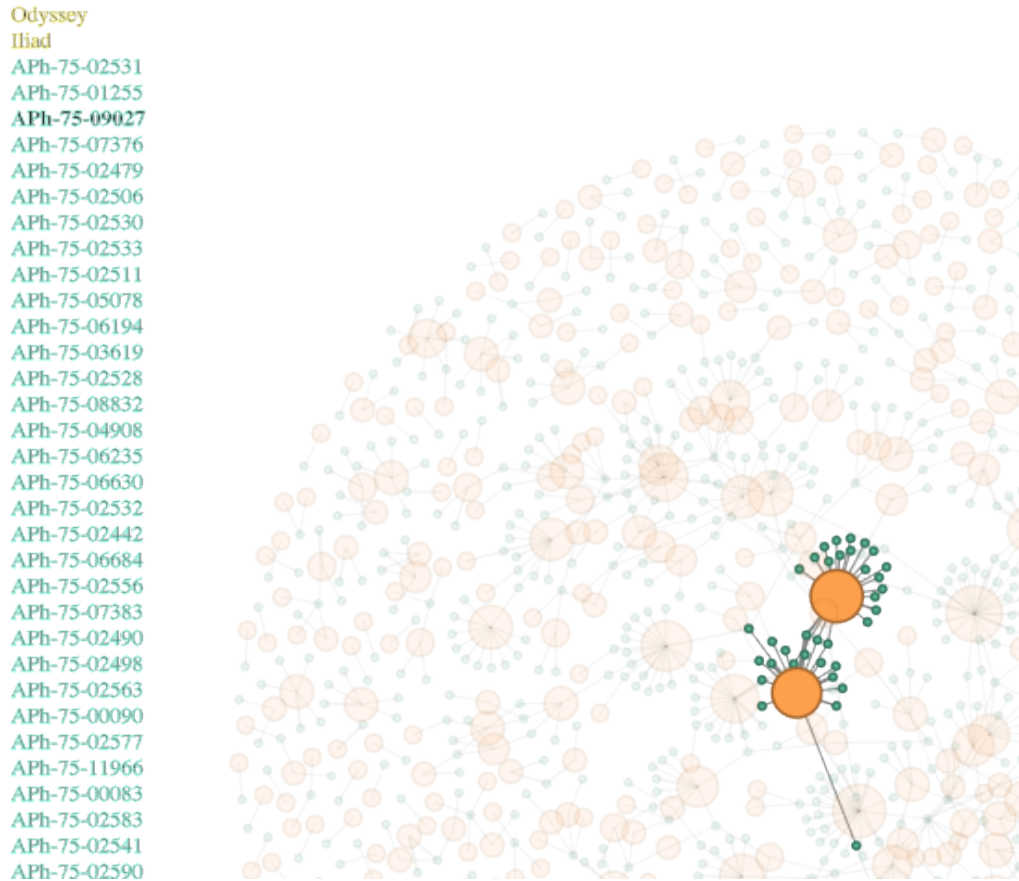


Figure 5.6: A screenshot of the interactive visualisation of the meso-level network extracted from the APh data. The highlighted nodes correspond to the results of a search for documents citing the *Odyssey* and the *Iliad*. The graph shows intuitively what are the abstracts (green nodes) that refer to both works (orange nodes).

that cite Vergil (see figure 5.7). Moreover, since the citation network has three different levels of granularity – macro, meso and micro – similar search patterns can be elicited at each level. In fact, it is possible to expand the search to include related works at the meso-level and related text passages at the micro-level.

The same principle can be applied to searches having as a starting point a given document within a collection. Suppose that the user has already found an abstract of interest. By searching for all the connected nodes within a distance of two hops it is possible to retrieve and suggest to the user related documents. The nature of the relatedness depends on the network level that is considered. If the macro-level network is searched, the results will favour recall over precision, meaning that the search returns also some potentially unrelated results. In fact, since at this level abstracts are connected when they refer to the same au-

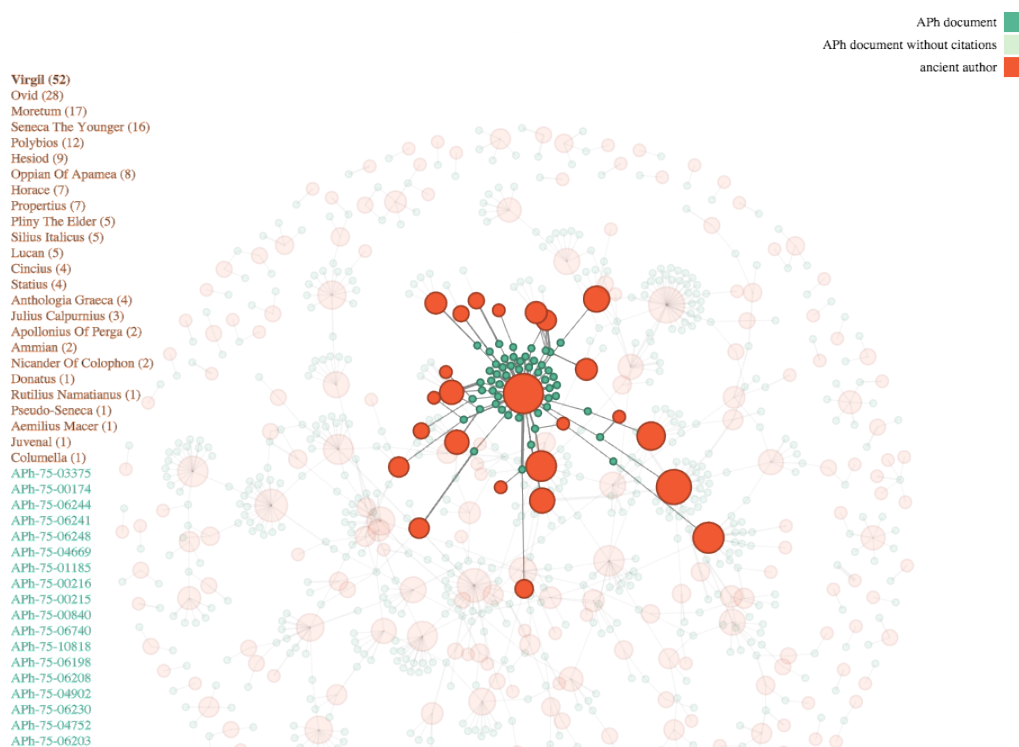


Figure 5.7: A screenshot of the interactive visualisation of the macro-level network extracted from the APh data. The highlighted nodes correspond to the abstracts citing Vergil (the red node in the middle). The search is expanded to include authors that are related as they are cited by abstracts citing Vergil.

thor the suggested results may only loosely be related to the start document. Conversely, if the micro-level network is searched, fewer but more closely related documents will be suggested. In fact, at the micro-level two documents are related when they share the references to one or more text passages.

5.3.3 *Advantages and Limitations*

As any lens does, the network lens implies some degree of distortion. Indeed, such a citation network puts textual references first – and thus textual matters – before any other aspect of the study of classical texts. As distortion and effectiveness of this lens are deeply intertwined they need to be teased apart so that the tradeoff inherent to this kind of searching is not forgotten.

The search that was just discussed will certainly prove useful to those who are concerned with the study of specific textual matters. Since a network focusses by nature on relations between entities, such a search

favours an intertextual perspective as it draws the scholar's attention to authors, works or passages that are discussed in relation to each other. Similarly, the ability that this network provides to retrieve publications that cite the same set of passages and are thus related is particularly crucial for those concerned with the study of intertextual parallels.

Although there is an inherent bias in this approach it is also what makes it effective. In the case of the search example above, the exclusive focus on canonical references that this network-lens implies allows for retrieving, in an effective way, those two publications on *Acharni-ans* 9–12 out of the 6,947 which appeared in 2004 and were reviewed in volume 75 of the APh. Since the search is driven by references to primary sources, those documents that do not contain any such reference are simply not taken into account. The advantage of such a strong assumption is that documents that are very likely to be irrelevant – as they do not cite nor mention any ancient texts – can be simply filtered out. The larger is the body of secondary literature being searched, the more valuable the advantages of such a selective search become.

The view on texts through this lens, however, seems to distort – or better flatten – two aspects of citations. First, this network provides no background as to why a given source is cited. There are many reasons for referring to a text passage such as to discuss an issue of syntax or style or to discuss an intertextual parallel between texts. Since the motivation underlying a reference is not captured, this characterisation is absent from the citation network. In fact, all references are represented in this citation network in the same way, namely as connections between nodes. In contrast, this aspect is captured – although not at the same level of granularity – in the APh where each reviewed publication is classified according to the specific aspects it focusses on.

The second aspect that is not taken into account is the importance of the cited text passage in relation to the publication as a whole. The only current characterisation of a citation is its frequency, which is expressed by the weight of edges in the network. Beyond frequency, however, all canonical references are treated equally, no matter if the cited text is the main focus of a publication or if the citation is only marginal to the main topic discussed.

This is less of a problem in the examples discussed in this section because the APh abstracts provide a summary of the main contributions of a publication. Therefore, the references that are found in the sum-

mary were already selected by the reviewer as being highly relevant to the publication reviewed. Nevertheless, this aspect is going to be more problematic, and will deserve further investigation, in cases where the citation network is constructed from the full text of journal articles. The challenge is how to determine the importance of a reference within an article that contains a hundred of them and whose argument may unfold over several pages. Additional indicators of the importance of a reference that may be taken into account are its presence in the main title (or abstract) of the publication, or the location in the text where is found (e.g. in the text body or in a footnote).

5.4 SUMMARY

The three-level citation network presented in this chapter enables us to represent explicitly the web of relations that canonical references create, whilst preserving the different hierarchical levels that these references imply. Once formalised as a network, this web of relations can be analysed, visualised and searched. The ability to search for specific text passages cited within publications is already offered in some form by existing *indices locorum*. However, a citation network enables new ways of searching that leverage the relations between publications which are created by citations. This network lens makes it possible to find effectively publications that cite the same set of authors, works or text passages. Searching this network, however, is only one of its possible uses. In the next chapter I consider the potential that lies in the analysis of citation networks extracted from large-scale bodies of secondary literature.

CONCLUSIONS

Overview

In section 6.1 I summarise the contributions of this dissertation to Digital Classics research. In section 6.2 I discuss ways in which the system I have devised to extract canonical references could be further developed. Section 6.3 considers the new research perspectives that this study opens up. Finally, section 6.3 concludes this chapter by reflecting on the implications of applying computing to research problems in the humanities.

6.1 OVERALL CONTRIBUTIONS OF THIS WORK

The work presented in this dissertation makes three main contributions to research in the area of Digital Classics:

1. an approach to the automatic extraction of canonical references;
2. a formal ontology of such references that allows us to publish the extracted citation data online by means of semantic technologies;
3. a three-level citation network laying the foundations for the development of new search tools.

6.1.1 *A System to Extract Canonical References*

This dissertation presented an approach to the automatic extraction of canonical references from texts (chapter 4). This approach consists of treating the extraction of these references as a problem of domain-specific Named Entity Recognition. Although this work does not contribute any new method or algorithm to research in this field, it demon-

strated the suitability of Named Entity Recognition as an overall framework for capturing these references automatically.

I have devised the extraction as a three-step process which aligns with the steps into which the task of extracting named entities is usually broken down (section 4.4). First, the citation components are extracted (named entity extraction); second, the relations between components are identified (relation extraction); third, the disambiguation of the extracted citations (entity and relation disambiguation). Ultimately, disambiguating a reference means transforming a human-readable string of characters into a machine-readable identifier – a CTS URN – that identifies the cited text passage.

I have implemented this approach as a working prototype and tested it against two datasets, the abstracts of *L'Année Philologique* and the Classics articles contained in JSTOR. Moreover, I have evaluated the overall performance of this prototype system as reported in section 4.5. To this purpose, a subset of *L'Année Philologique* (APh) abstracts was processed and then manually corrected in order to measure the accuracy of each of three steps of the extraction process.¹

The system performs generally well in the case of highly structured citations (e.g. Verg., *Aen.* 1.1–10). In contrast, the greatest difficulties are encountered in those cases where the author elides some details, relying on the ability of the trained human reader to derive them from the surrounding context. It should be noted that this tendency to elide details of a reference may be intensified, in the case of the texts that were used for the evaluation, by their concise nature. Nevertheless, the difficulties encountered highlight that a computational understanding of context remains one of the greatest challenges in the disambiguation of canonical references and, more generally, in the automatic processing of natural language.² In some cases found in the APh, it is the title of the reviewed publication that determines the context necessary to disambiguate the references contained in the abstract (section 4.5). In other cases, the context is provided by the references contained in

¹ The source code of the prototype as well as the manually corrected data have been made available to the community under an open source licence in the hope that they will spark further research and development in this area. The code and the data can be found respectively at <http://dx.doi.org/10.5281/zenodo.10886> and <http://dx.doi.org/10.5281/zenodo.12762>.

² See *infra* at p. 161.

the previous sentences and is necessary to resolve a highly ambiguous abbreviation (e.g. “*Epist.*”).

6.1.2 *An Ontology of Canonical References*

The second contribution to research is constituted by Humanities Citation Ontology (HuCit), a formal ontology of canonical references (chapter 3). This ontology formalises the underlying model that the practice of citing texts already implies and reflects. The rationale behind its design is to foster the interoperability and reuse of data in two ways: first, by relying on high level ontologies such as CIDOC CRM and FRBR_{OO}; and, second, by aligning fully with the CTS protocol – the *de facto* standard within the Digital Classics community to express canonical references in a machine actionable way.³

The main advantage of this ontology lies in that it allows for instantiating any citable passage within the canonical scheme of a given text. Once instantiated, a passage can then be assigned an HTTP URI, thus becoming “citable” within other RDF statements published on the Web. In other words, this ontology makes it possible to formulate machine-understandable statements – in the form of RDF triples – about specific sections of a canonical text.

A possible application of HuCit is in a *semantic publishing* context. For example, the articles of an online Classics journal could be enriched by encoding the canonical references they contain as links to instances of HuCit, thus enhancing their discoverability.

6.1.3 *A Citation-based Search*

The third main contribution of this work is to enable the development of new tools for searching through Classics publications by the references they contain (chapter 5). The ability to extract automatically canonical references constitutes a substantial change in the scale of publications that can be indexed – and thus made searchable – as compared with manually created *indices locorum*. The three-level network that I have devised to represent the extracted web of citations enables a number of new search functionalities that leverage the properties of the underlying

³ In fact, HuCit follows so closely the CTS model that it can almost be considered a formalisation of its underlying conceptual model.

network (section 5.3). For example, it allows for clustering publications on the basis of the primary sources they cite. These clusters of related publications could then be used as a basis to provide a recommendation feature – a functionality that many bibliographic indexes already offer. The innovative aspect of this feature lies in the ability to recommend publications that are related as they cite the same set of primary sources.

Furthermore, designing a search interface that allows classicists to explore intuitively this three-level network presents some interesting challenges. One challenge is to enable the user to move back and forth between the three levels of the citation network (macro-, meso- and micro-level). Another challenge is to find a visual metaphor that combines the advantages of a graph visualisation with the familiarity of having a tree-like index of cited passages. The advantage of visualising citations between documents as a graph lies in that it facilitates the activity of browsing through connections, which can lead to the serendipitous discovery of relevant information. An index that allows multiple passages to be selected, however, would enable the user to see immediately what can be searched for. All these aspects of designing a search interface tailored to the classicists' needs constitute interesting areas for further research.

6.2 FUTURE DEVELOPMENT

Among the contributions to research made by this dissertation, the system for the automatic extraction of canonical references is certainly the one with the greatest potential for further development. It is possible to identify three main directions for this development:

1. improving the overall accuracy of the system;
2. extending the extraction to other kinds of references;
3. making the system available in the form of a web service as part of a digital research infrastructure.

6.2.1 *Improving the Accuracy of Canonical Reference Extraction*

Since the outset, the focus of this research has been on investigating the feasibility of the automatic extraction of canonical references rather than

implementing a system that achieves the best possible performance. As a result, only a limited number of approaches were tested for each of the three extraction steps.

In particular, it was only possible to apply a machine learning-based approach to the extraction and classification of named entities (section 4.5.1). Its application to the remaining two processing steps – relation extraction and disambiguation – where a rule-based approach was used is certainly possible, yet remains to be explored and evaluated. The main advantage of this approach lies in the ability to train the entire system to better cope with the characteristics of a specific set of materials, thus improving the overall accuracy of the results.

Another key to achieve better results in the disambiguation of references, as shown by the evaluation, is to have a more comprehensive *knowledge base* of canonical citation schemes, names of authors, titles of works and abbreviations (section 4.5). The knowledge base created as part of this research is certainly a starting point, yet is far from being comprehensive. One possible way to increase its coverage would be to feed the results of the automatic extraction back into the knowledge base. For example, new abbreviations that are correctly captured by the system could be added back to it. Since this operation requires some manual checking, it could be combined with the manual correction necessary to produce training data for the machine learning-based components of the system.

6.2.2 *Extending the Extraction to other Kinds of References*

This work focussed on one specific kind of textual references. However, Classics publications refer to many other kinds of sources such as fragmentary texts, manuscripts, inscriptions and papyri in addition to bibliographic references to other publications. Moreover, if we extend the disciplinary scope to include Classical Archaeology, references to coins, vases and other material objects are also found. All this richness of references could also be captured automatically by extending the functionalities of the citation extraction system.

Although single components of the system would need to be tailored to the characteristics of the chosen references, two aspects of the approach I have proposed in this dissertation are general enough to be applicable in this context. First, the use of a knowledge base to provide

the extraction system with the domain-specific knowledge necessary to capture and interpret the references (e.g. abbreviations, name variants, etc.). Second, the linking of these references to the referred object – be it an inscription, papyrus or a coin – by means of HTTP URIs. This is made possible by the existence of openly available resources for each of these types of objects that publish their data following the approach.⁴

Taking into account these additional types of references would allow us to better represent the complex network of relations that characterises classical scholarship. Indeed, the citation network currently takes into account only canonical references and therefore represents a simplified view of a much richer and more complex system of connections between objects (section 5.3.1). Moreover, having such an extended network would allow us to refine further the concept of relatedness between publications. Referring to the same text passage would no longer be the only criterion but just one among several criteria that make up the relatedness between two or more publications.

6.2.3 *Citation Extraction as Research Infrastructure*

One aspect that emerged from this research is that the extraction of references from large-scale resources such as JSTOR requires the computational power that a research infrastructure can offer.⁵ Projects such as DARIAH and CLARIN are currently setting up such research infrastructure for scholars in the Arts and Humanities across Europe. The benefit of having the reference extraction provided as part of this infrastructure is two-fold. First, it makes this service available to a wider community of scholars. Second, it constitutes a more robust solution for the mining of large-scale resources.

⁴ Examples of such resources are Papyri.info <http://papyri.info/>, Eagle <http://www.eagle-network.eu/> as a central aggregator of epigraphic data, Online Coins of the Roman Empire <http://numismatics.org/ocre/> and Arachne <http://arachne.dainst.org/> for archaeological objects.

⁵ I have estimated that it would take approximately more than 3 months (of computing time) on a laptop machine to extract canonical references from the approximately 110,000 Classics journal articles contained in JSTOR.

6.3 FUTURE USE AND IMPLICATIONS

In addition to areas for further development, this research has also highlighted other potential areas for the application of the work presented in this dissertation. These areas include:

1. the consumption of automatically extracted references;
2. the mining of other types of publications such as classical commentaries;
3. the analysis and visualisation of the extracted citation networks.

6.3.1 *Enhancing the Reading of Classical Texts*

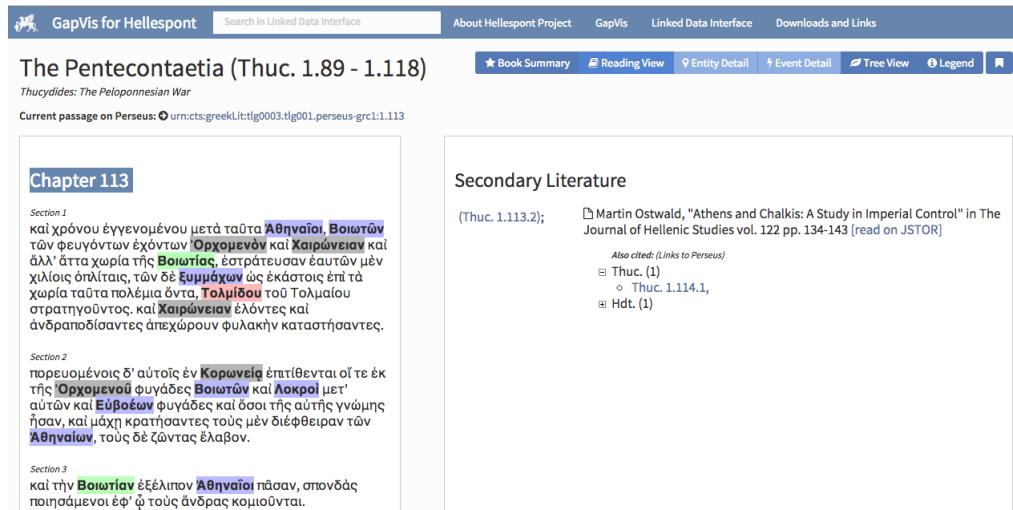
Publishing online the automatically extracted references as semantic data enables the consumption and re-use of this data in various contexts. Since HuCit makes use of CTS URNs to identify the cited text passages, other resources that rely on these identifiers can consume this data for other purposes such as to enhance the reading of classical texts.

A chance to experiment with this potential use of citation data was provided by the Hellespont project.⁶ The project investigated how several layers of annotation can be brought together into a single environment to enhance our reading of one section of Thucydides' account of the Peloponnesian war, the so-called "Pentecontaetia" (Thuc. 1.89–1.118).

The annotations which enriched the text included the linguistic annotation of morphological and syntactical features as well as the manual annotation of named entities and temporal events within the text. Additionally, the system presented in this dissertation was employed to mine citations to the "Pentecontaetia" from secondary literature contained in JSTOR (Romanello and Thomas, 2012). Links to the journal articles citing the current passage being read are displayed to the reader in one dedicated view of the Hellespont reading environment (see figure 6.1).

This function of referring to secondary literature relevant to the interpretation of a given passage is already performed by classical commentaries. However, two main differences are worth noting. First, the related literature which is brought to the attention of the reader is not

⁶ The Hellespont Project, <http://hellespont.dainst.org/>.



GapVis for Hellespont Search in Linked Data Interface About Hellespont Project GapVis Linked Data Interface Downloads and Links

The Pentecontaetia (Thuc. 1.89 - 1.118)
Thucydides: The Peloponnesian War
Current passage on Perseus: urn:cts:greekLit:tlg003.tlg001.perseus-grc1:1.113

Chapter 113

Section 1
καὶ χρόνου ἐγγενομένου μετὰ ταῦτα Ἀθηναῖοι, Βοιωτῶν τῶν φευγόντων ἐχόντων Ὀρχομενὸν καὶ Χαιρώνειαν καὶ ἄλλ' ἅττα χωρία τῆς Βοιωτίας, ἐστράτευσαν ἑαυτῶν μὲν χίλιοις ὀπλίταις, τῶν δὲ Συμμάχων ὡς ἐκάστοις ἐπὶ τὰ χωρία ταῦτα πολέμια ὄντα, Τολμίδου τοῦ Τολμαίου στρατηγοῦντος, καὶ Χαιρώνειαν ἐλόντες καὶ ἀνδραποδίσαντες ἀπεχώρουν φυλακὴν καταστήσαντες.

Section 2
πορευομένοις δ' αὐτοῖς ἐν Κορωνείᾳ ἐπιτίθενται οἱ τε ἐκ τῆς Ὀρχομενοῦ φυγάδες Βοιωτῶν καὶ Ἀοκροὶ μετ' αὐτῶν καὶ Εὐβοέων φυγάδες καὶ ὅσοι τῆς αὐτῆς γνῶμης ἦσαν, καὶ μάχῃ κρατήσαντες τοὺς μὲν διέφθειραν τῶν Ἀθηναίων, τοὺς δὲ ζώντας ἔλαβον.

Section 3
καὶ τὴν Βοιωτίαν ἐξέλιπον Ἀθηναῖοι πᾶσαν, σπονδὰς ποιησάμενοι ἐφ' ᾧ τοὺς ἀνδρας κομιοῦνται.

Secondary Literature

(Thuc. 1.113.2); Martin Ostwald, "Athens and Chalkis: A Study in Imperial Control" in The Journal of Hellenic Studies vol. 122 pp. 134-143 [read on JSTOR]

Also cited: (Links to Perseus)

- Thuc. (1)
 - Thuc. 1.114.1,
- Hdt. (1)

Figure 6.1: A screenshot of the secondary literature view in the Hellespont reading environment. The Greek text of the passage in focus is displayed in the panel on the left, while the articles extracted from JSTOR and related to this passage are shown on the right.

based on the selection made by the commentator. Instead, it contains all the journals in JSTOR citing the passage in focus, with the resulting need for the reader to assess the relevance of the results returned. Second, the digital environment potentially allows for displaying more information about the context where the citation is found, thus enabling the reader to discern more promptly its specific relevance to the passage being read. These two differences would need to be addressed when designing similar interfaces and reading environments. In particular, how to present a large number of related articles to the reader in an insightful way would need careful consideration.

6.3.2 Mining Intertextual Parallels from Commentaries

Although classical commentaries were not mined for citations in the context of this research, the wealth of intertextual parallels they contain constitute a great potential for further research (section 2.1). In particular, the extraction of these parallels from commentaries by using the system I have developed would enable a quantitative investigation of the history of this genre of publications.

Indeed, a central aspect of the contemporary debate on the status of the commentary concerns the choices made by commentators with regards to the selection of parallels that elucidate the text (section 2.1).

These choices concern the number of parallels that are included – the tendency of the commentator to be more or less selective – but also the extent to which the choice of parallels is innovative or conservative as compared with previous commentaries on the same text.

Observing how these two aspects vary over time could allow us to identify some tendencies in the history of commentary. By considering a diachronic selection of commentaries on one specific text, it may be possible to observe variations in the set of parallels used to elucidate the commented text. Questions that we could try to answer with this data at hand include: When did a given intertextual parallel first appear? Which parallels became “traditional” as they can be found consistently throughout commentaries on a given text? Is it possible to identify the commentators that innovated the most with regards to the choice of intertextual parallels? Moreover, the ability to quantify the frequency of parallels and its variation over time would allow us to test the hypothesis that electronic concordances have had an effect on how commentators select parallel passages (section 2.1.5).

Some challenges would need to be tackled, however, to enable such a study of commentaries. Capturing the canonical references that signal parallel passages can be accomplished automatically – to some degree – by using the system presented in this dissertation. Conversely, determining the *lemma* to which the parallels are related would inevitably require manual encoding. Indeed, commentaries are characterised by a sophisticated structure and a complex system of references to the commented text that are hard to capture automatically. A further challenge is constituted by the alignment of the various *lemmata* on a common text. Since different commentaries may refer to different critical editions of the commented text, a *lemma* in a given commentary may contain a variant or conjecture that is absent from other editions. Therefore, such an alignment is necessary in order to be able to compare these *lemmata*.

Furthermore, in addition to enabling a quantitative study of the history of commentaries, the intertextual parallels extracted from commentaries could be used as a benchmark set for evaluating the automatic detection of allusions. This approach was developed to evaluate the tool Tesserae. A set of parallels collated from commentaries was used to assess the accuracy with which the tool identifies automatically intertextual parallels between two texts.⁷

⁷ See *infra* at p. 47.

6.3.3 *Analysis and Visualisation of the Citation Network*

The analysis of the three-level citation network, which was discussed in chapter 5 with regards to searching, constitutes another promising area for further research. Such a network could be constructed by mining canonical references from a large-scale archive such as JSTOR, which has the advantage of covering a wide timespan, thus allowing for longitudinal – i.e. diachronic – analysis of the network.

The main assumption for the study of this network is that canonical references represent the traces of the debated topics that scholars leave in their publications. Under this assumption, the varying frequency of citations to a given author can be seen as a proxy for the attention received by that author in different periods. Given the three levels of this network – macro-, meso- and micro-level – similar analysis are possible for individual works as well as specific text passages.

Moreover, since networks by their very nature focus on the relations between the entities they represent, the analysis of this network has the potential to highlight some trends as to how texts were studied in relation to each other. A question that could be investigated by using this approach is to what extent it is possible to identify clusters of authors and how these clusters evolved over time. Since the majority of clustering algorithms are implemented specifically for one-mode networks, one possible way to apply such algorithms is to project the two-mode networks into several one-mode, undirected networks. For example, the macro-level citation network could be projected into an author-author network – where two authors are connected when they are cited by the same article – as well as into an article-article network – where two articles are connected when they cite the same author. The former can be considered as a reflection of the literary system, whilst the latter could allow us to identify clusters of related articles.

6.4 FINAL REFLECTIONS

The process of translating the act of decoding citations into computational terms has drawn my attention to previously unconsidered aspects of citing. As classicists we constantly deal with canonical references and citation schemes, but we rarely have the chance to reflect on their impli-

cations: their nature of social and historical constructs and how these schemes evolve at every transition to a new medium.

It is also easy to forget, as readers, the assumed knowledge that we have to supply for example to decode a reference to an author's *opus maximum*. In contrast, devising an automatic system to decode such references revealed the amount of implicit knowledge about ancient authors and texts that we need to formalise and to provide a computer programme with.

This self-reflection, prompted by the use of computing, has been a characteristic of digital humanities research since its very beginning. Consider what Busa wrote some years ago with regards to the *Index Thomisticus*:

In fact, the computer has even improved the quality of methods in philological analysis, because its brute physical rigidity demands full accuracy, full completeness, full systematicity. Using computers I had to realize that our previous knowledge of human language was too often incomplete and anyway not sufficient for a computer program. Using computers will therefore lead us to a more profound and systematic knowledge of human expression; in principle, it can help us to be more humanistic than before (Busa, 1980, p. 89).

It is the application of computing to humanities problems that, ultimately, forces us to such reflections, thus leading us potentially to a more profound and systematic knowledge of our objects of study as well as of our scholarly practices.

GLOSSARY

Knowledge Base	A database containing a set of instances of the classes defined in the underlying <i>ontology</i> . A knowledge base is often employed in the context of an information extraction system as a surrogate of knowledge about a given domain. 52, 62, 63, 65, 78, 97, 100–109, 117, 118, 125, 136, 144, 146, 150, 157, 158, 160–162, 164, 191, 198
Linked Open Data	A set of best practices for publishing semantic data on the Web. It advocates the use of HTTP Uniform Resource Identifiers (URIs) as names for things and the use of links between resources and vocabularies as a way of integrating information in a machine-understandable way. 11, 66–68, 76, 83, 84, 100, 101
Ontology	A computational artefact which models a domain of interest by defining classes of objects, their attributes and the relations between them. 43, 44, 50, 54, 60–62, 109, 198
Opus Maximum	The only work produced by a given author or, in the case of multiple works, the most known. When citing an opus maximum the indication of the cited work can be omitted (e.g. Prop. 2.22). 52, 99, 107, 122, 123, 197, 198
Semantic Publishing	The activity of enriching the content of publications by means of metadata expressed using semantic web standards. These metadata make it possible to discover and integrate more easily and effectively information contained within publications. 69, 73, 189

Text reuse The meaningful reiteration of text, usually beyond the mere repetition of common language. The term *text reuse* encompasses a number of different relations between texts such as translation, quotation, allusion and plagiarism. 46, 47, 199

BIBLIOGRAPHY

- Almas, B., Babeu, A., and Krohn, A. (2014). Linked Data in the Perseus Digital Library. *ISAW Papers*, 7(3).
- Almas, B. and Berti, M. (2013). Perseids Collaborative Platform for Annotating Text Re-Uses of Fragmentary Authors. In Tomasi, F. and Vitali, F., editors, *DH-Case 2013*.
- Anderson, S., Blanke, T., and Dunn, S. (2010). Methodological commons: arts and humanities e-Science fundamentals. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1925):3779–3796.
- Babeu, A. (2008). Building a “FRBR-Inspired” Catalog: The Perseus Digital Library Experience. <http://www.perseus.tufts.edu/publications/PerseusFRBRExperiment.pdf>.
- Babeu, A. (2011). Rome Wasn’t Digitized in a Day: Building a Cyber-infrastructure for Digital Classics. Technical Report (Draft Version 1.2), Perseus Digital Library, Council on Library and Information Resources.
- Bamman, D. and Crane, G. (2008). The logic and discovery of textual allusion. In Ribarov, K. and Sporleder, C., editors, *In: Proceedings of the Second Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2008)*, Marrakesh.
- Bamman, D. and Crane, G. (2009). Discovering Multilingual Text Reuse in Literary Texts. <http://www.perseus.tufts.edu/publications/2009-Bamman.pdf>.
- Barnes, J. (1995). *The Cambridge companion to Aristotle*. Cambridge University Press, Cambridge; New York.
- Barnet, B. (2013). *Memory machines : the evolution of hypertext*. Anthem Press, London.

- Bates, M. J. (1989). The design of browsing and berrypicking techniques for the online search interface. *Online Information Review*, 13(5):407–424.
- Bates, M. J. (1996). The Getty End-User Online Searching Project in the Humanities: Report No. 6: Overview and Conclusions. *College & Research Libraries*, 57(6):514–523.
- Benardou, A., Constantopoulos, P., and Dallas, C. (2013). An Approach to Analyzing Working Practices of Research Communities in the Humanities. *International Journal of Humanities and Arts Computing*, 7(1-2):105–127.
- Benardou, A., Constantopoulos, P., Dallas, C., and Gavrilis, D. (2010). Understanding the Information Requirements of Arts and Humanities Scholarship. *International Journal of Digital Curation*, 5(1):18–33.
- Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5):34–43.
- Berti, M. (2013). Collecting Quotations by Topic: Degrees of Preservation and Transtextual Relations among Genres. *Ancient Society*, 43:269–288.
- Blackwell, C. and Smith, N. (2012). An overview of the CTS URN notation. <http://www.homermultitext.org/hmt-doc/cite/cts-urn-overview.html>.
- Blanke, T. and Hedges, M. (2013). Scholarly primitives: Building institutional infrastructure for humanities e-Science. *Future Generation Computer Systems*, 29(2):654–661.
- Bolter, J. D. (1991). The Computer, Hypertext, and Classical Studies. *The American Journal of Philology*, 112(4):541–545.
- Borthwick, A., Sterling, J., Agichtein, E., and Grishman, R. (1998). Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition. In *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 152–160.
- Boschetti, F. (2007). Methods to extend Greek and Latin corpora with variants and conjectures: Mapping critical apparatuses onto reference text. In *Proceedings of the Corpus Linguistics Conference 2007*.

- Briggs, W. W. and Calder, W. M. (1990). *Classical scholarship: a biographical encyclopedia*. Garland Pub., New York.
- Brughmans, T. (2013). Networks of networks: a citation network analysis of the adoption, use, and adaptation of formal network techniques in archaeology. *Literary and Linguistic Computing*, 28(4):538–562.
- Brunner, T. F. (1993). Classics and the Computer: The History of a Relationship. In *Accessing antiquity : the computerization of classical studies*, pages 10–33. University of Arizona Press, Tucson.
- Buchanan, G., Cunningham, S. J., Blandford, A., Rimmer, J., and Warwick, C. (2005). Information Seeking by Humanities Scholars. In Rauber, A., Christodoulakis, S., and Tjoa, A. M., editors, *Research and Advanced Technology for Digital Libraries*, number 3652 in Lecture Notes in Computer Science, pages 218–229. Springer Berlin Heidelberg.
- Büchler, M. (2013). *Informationstechnische Aspekte des Historical Text Re-use*. PhD thesis, Universität Leipzig.
- Büchler, M., Geßner, A., Berti, M., and Eckart, T. (2012). *Measuring the Influence of a Work by Text Re-Use*, volume The Digita, pages 63–79.
- Buitelaar, P. and Magnini, B. (2005). Ontology Learning from Text: An Overview. In Buitelaar, P., Cimiano, P., and Magnini, B., editors, *Ontology Learning from Text: Methods, Applications and Evaluation*, pages 3–12. IOS Press.
- Busa, R. (1980). The annals of humanities computing: The index Thomisticus. *Computers and the Humanities*, 14(2):83–90.
- Byrne, K. and Klein, E. (2010). Automatic Extraction of Archaeological Events from Text. In Frischer, B., editor, *Making history interactive: computer applications and quantitative methods in archaeology (CAA) ; proceedings of the 37th international conference, Williamsburg, Virginia, United States of America, March 22 - 26, 2009*, volume 2079 of *BAR International Series*, pages 48–56. Archaeopress, Oxford u.a.
- Calame, C. (2001). Le scienze dell’Antichità tra neoliberalismo e cultura da supermercato: inflazione bibliografica e smarrimento metodologico. *I Quaderni del Ramo d’Oro*, 4:181–203.

- Cerri, G. (2009). Inflazione bibliografica e mutamento antropologico degli studiosi di antichistica. *Lexis*, 27:253–261.
- Chieu, H. L. and Ng, H. T. (2003). Named Entity Recognition with a Maximum Entropy Approach. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4, CONLL '03*, pages 160–163, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Ciccarese, P., Shotton, D., Peroni, S., and Clark, T. (2014). CiTO + SWAN: The web semantics of bibliographic records, citations, evidence and discourse relationships. *Semantic Web*, 5(4):295–311.
- Ciccarese, P., Wu, E., Wong, G., Ocana, M., Kinoshita, J., Ruttenberg, A., and Clark, T. (2008). The SWAN biomedical discourse ontology. *Journal of Biomedical Informatics*, 41(5):739–751.
- Cimiano, P. (2006). *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer, Berlin.
- Cimiano, P., Völker, J., and Studer, R. (2006). Ontologies on Demand? - A Description of the State-of-the-Art, Applications, Challenges and Trends for Ontology Learning from Text. *Information, Wissenschaft und Praxis*, 57(6-7):315–320.
- Coffee, N. (2014). Response of Christopher Forstall. <http://tesseract.caset.buffalo.edu/blog/response-of-christopher-forstall/>.
- Coffee, N., Koenig, J.-P., Poornima, S., Forstall, C. W., Ossewaarde, R., and Jacobson, S. L. (2013). The Tesseract Project: intertextual analysis of Latin poetry. *Literary and Linguistic Computing*, 28(2):221–228.
- Cohen, K., Demner-Fushman, D., Ananiadou, S., and Tsujii, J.-i., editors (2014). *Proceedings of BioNLP 2014*. Association for Computational Linguistics, Baltimore, Maryland.
- Cohen, K. B., Demner-Fushman, D., Ananiadou, S., Pestian, J., and Tsujii, J., editors (2013). *Proceedings of the 2013 Workshop on Biomedical Natural Language Processing*. Association for Computational Linguistics, Sofia, Bulgaria.

- Connor, W. R. (1990). Scholarship and Technology in Classical Studies. In Katzen, M., editor, *Scholarship and Technology in the Humanities*, pages 52–62.
- Conte, G. B. (1974). *Memoria dei poeti e sistema letterario: Catullo, Virgilio, Ovidio, Lucano*. G. Einaudi, Torino.
- Councill, I. G., Giles, C. L., and Kan, M.-y. (2008). ParsCit: An open-source CRF Reference String Parsing Package. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odjik, J., Piperidis, S., and Tapias, D., editors, *Proceedings of LREC*, number 3, pages 661–667. Citeseer, European Language Resources Association (ELRA).
- Cowan, R. (2013). Of Gods, Men and Stout Fellows: Cicero on Sallustius' Empedoclea (Q. Fr. 2.10[9].3). *The Classical Quarterly (New Series)*, 63(02):764–771.
- Cozzo, A. (2006). *La tribù degli antichisti : un'etnografia ad opera di un suo membro*. Carocci, Roma.
- Crane, G. (2004). Classics and the computer: an end of the history. In *A Companion to Digital Humanities*, pages 46–55.
- Crane, G., Almas, B., Babeu, A., Cerrato, L., Krohn, A., Baumgart, F., Berti, M., Franzini, G., and Stoyanova, S. (2014). Cataloging for a Billion Word Library of Greek and Latin. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage, DATeCH '14*, pages 83–88, New York, NY, USA. ACM.
- Crane, G., Finnegan, R., Glock, A. E., Hirsch, E., Martindale, C., Stoddart, S., Michel, J.-H., Morris, I., and Probst, P. (1991). Composing Culture: The Authority of an Electronic Text [and Comments and Reply]. *Current Anthropology*, 32(3):293–311.
- Crane, G., Seales, B., and Terras, M. (2009). Cyberinfrastructure for Classical Philology. *Digital Humanities Quarterly*, 3(1).
- David Mimno, G. C. (2005). Hierarchical Catalog Records: Implementing a FRBR Catalog. *D-lib Magazine - DLIB*, 11(10).
- de la Cerda, J. L. (1612). *P. Virgilii Maronis priores sex libri Aeneidos argumentis, explicationibus notis illustrati*. sumptibus Horatij Cardon, Lugduni.

- Doerr, M. and Iorizzo, D. (2008). The dream of a global knowledge network - A new approach. *Journal on Computing and Cultural Heritage*, 1(1):1–23.
- Doerr, M. and LeBoeuf, P. (2007). Modelling Intellectual Processes: The FRBR - CRM Harmonization. In Thanos, C., Borri, F., and Candela, L., editors, *Digital Libraries: Research and Development*, volume 4877 of *Lecture Notes in Computer Science*, pages 114–123. Springer Berlin / Heidelberg.
- Easley, D. and Kleinberg, J. (2010). *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, volume 1. Cambridge University Press.
- Ekbal, A., Bonin, F., Saha, S., Stemle, E., Barbu, E., Cavulli, F., Girardi, C., and Poesio, M. (2011). Rapid Adaptation of NE Resolvers for Humanities Domains using Active Annotation. *Journal for Language Technology and Computational Linguistics*, 26(2):39–51.
- Elliott, T. and Gillies, S. (2009). Digital Geography and Classics. *Digital Humanities Quarterly*, 3(1).
- Elliott, T., Heath, S., and Muccigrosso, J. (2014). Prologue and Introduction. *ISAW Papers*, 7(1).
- Ellis, D. (1989). A behavioural model for information retrieval system design. *Journal of Information Science*, 15(4-5):237–247.
- Febvre, L. and Martin, H.-J. (1958). *L'apparition du livre*. Éditions A. Michel, Paris.
- Feigenbaum, E. A. and Klah, P. (2003). Expert Systems. In *Encyclopedia of Computer Science*, pages 684–689. John Wiley and Sons Ltd., 4th edition.
- Fowler, D. (1997). On the Shoulders of Giants: Intertextuality and Classical Studies. *Materiali e discussioni per l'analisi dei testi classici*, (39):pp. 13–34.
- Fowler, D. (1999). Criticism as commentary and commentary as criticism in the age of electronic media. In Most, G. W., editor, *Commentaries = Kommentare*, number 4 in *Aporemata: Kritische Studien zur Philologiegeschichte*. Vandenhoeck & Ruprecht, Göttingen.

- Fraenkel, E. (1950). *Aeschylus Agamemnon*, volume I-III. Clarendon Press, Oxford.
- Franceschet, M. (2011). Collaboration in computer science: {A} network science approach. *Journal of the American Society for Information Science and Technology*, 62(10):1992–2012.
- Franceschet, M. (2012). The Large-Scale Structure of Journal Citation Networks. *Journal of the American Society for Information Science and Technology*, 63(4):837–842.
- Francesconi, E., Montemagni, S., Peters, W., and Tiscornia, D., editors (2010). *Semantic Processing of Legal Texts - Where the Language of Law Meets the Law of Language*.
- Galibert, O., Rosset, S., Tannier, X., and Grandry, F. (2010). Hybrid Citation Extraction from Patents. In Chair, N. C. C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation LREC10*. European Language Resources Association (ELRA).
- Garfield, E. (1980). Is Information Retrieval in the Arts and Humanities Inherently Different from That in Science? The Effect That ISI. *Library Quarterly*, 50(1):40–57.
- Genesereth, M. R. and Nilsson, N. J. (1987). *Logical foundations of artificial intelligence*. Morgan Kaufmann, San Francisco.
- Giaretta, P. and Guarino, N. (1995). Ontologies and Knowledge Bases: Towards a Terminological Clarification. In *Towards Very Large Knowledge Bases*, volume 1, pages 25–32. IOS Press, Amsterdam.
- Gibson, R. (2002). Cf. e.g.: a typology of ‘parallels’ and the function of commentaries on Latin poetry. In Gibson, R. K. and Kraus, C. S., editors, *The classical commentary : histories, practices, theory*, pages 331–357.
- Giles, C. L., Bollacker, K. D., and Lawrence, S. (1998). CiteSeer: An Automatic Citation Indexing System. In *Proceedings of the Third ACM Conference on Digital Libraries, DL '98*, pages 89–98, New York, NY, USA. ACM.

- Gioseffi, M. (2008). Come nasce un commento?: La formula "id est.". *Voces*, (19):71–92.
- Goldhill, S. (1999). Wipe Your Glosses. In Most, G. W., editor, *Commentaries = Kommentare*, pages 380–425.
- Grafton, A., Most, G. W., and Settis, S., editors (2010). *The Classical Tradition*. The Belknap Press of Harvard University Press, Cambridge, MA, USA & London, England.
- Grisham, R. (2010). Information Extraction. In Clark, A., Fox, C., and Lappin, S., editors, *The handbook of computational linguistics and natural language processing*. Wiley-Blackwell, Chichester, West Sussex; Malden, MA.
- Grover, C., Tobin, R., Byrne, K., Woollard, M., Reid, J., Dunn, S., and Ball, J. (2010). Use of the Edinburgh geoparser for georeferencing digitized historical collections. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1925):3875–3889.
- Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, 43(5–6):907–928.
- Guarino, N., Oberle, D., and Staab, S. (2009). What Is an Ontology? In *Handbook on Ontologies*, pages 1–17.
- Guarino, N. and Welty, C. A. (2009). An Overview of OntoClean. In *Handbook on Ontologies*, pages 201–220.
- Hardwick, L. (2000). Electrifying the Canon: The Impact of Computing on Classical Studies. *Computers and the Humanities*, 34(3):279–295.
- Hastie, T., Tibshirani, R., and Friedman, J. (2009). The Elements of Statistical Learning. *Elements*, 1:337–387.
- Hauptman, R. (2008). *Documentation: a history and critique of attribution, commentary, glosses, marginalia, notes, bibliographies, works-cited lists, and citation indexing and analysis*. McFarland, Jefferson, N.C.
- Heath, T. and Bizer, C. (2011). *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers.

- Hellqvist, B. (2010). Referencing in the Humanities and its Implications. *Journal of the American Society for Information Science and Technology*, 61(2):310–318.
- Heyworth, S. J. (2007). *Cynthia: a companion to the text of Propertius*. Oxford University Press, Oxford; New York.
- Higbie, C. (2010). Divide and Edit: a Brief History of Book Divisions. *Harvard Studies in Classical Philology*, 105:1–31.
- Hirst, G. (1997). Context as a Spurious Concept.
- Ireland, S. (1976). The Computer and Its Role in Classical Research. *Greece & Rome*, 23(1):40–54.
- Isaksen, L., Barker, E., Kansa, E. C., and Byrne, K. (2012). GAP: A Neogeo Approach to Classical Resources. *Leonardo*, 45(1):82–83.
- Jebb, R. C. (1894). *Sophocles: The Plays and Fragments*, volume 6. The University press.
- Jehasse, J. (1976). *La Renaissance de la critique: l'essor de l'humanisme érudit de 1560 à 1614*. Publications de l'Université de Saint-Étienne, Saint-Étienne.
- Jockers, M. L. (2013). *Macroanalysis: digital methods and literary history*. University of Illinois Press, Urbana; Chicago; Springfield.
- Jordanous, A., Lawrence, K. F., Hedges, M., and Tupman, C. (2012a). Exploring Manuscripts: Sharing Ancient Wisdoms Across the Semantic Web. In *Proceedings of the 2Nd International Conference on Web Intelligence, Mining and Semantics*, pages 44:1–44:12, New York, NY, USA. ACM.
- Jordanous, A., Stanley, A., and Tupman, C. (2012b). Contemporary transformation of ancient documents for recording and retrieving maximum information: when one form of markup is not enough. In *Proceedings of Balisage: The Markup Conference 2012*.
- Jurafsky, D. and Martin, J. H. (2009). *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Prentice Hall, Upper Saddle River, N.J.; London.

- Kalvesmaki, J. (2014). Canonical References in Electronic Texts: Rationale and Best Practices. *Digital Humanities Quarterly*, 8(2).
- Kecskeméti, J., Boudou, B., Cazes, H., and Céard, J., editors (2003). *La France des humanistes: Henri II Estienne, éditeur et écrivain*. Brepols, Turnhout.
- Lachmann, K. (1850). *Caroli Lachmanni in T. Lucretii Cari De rerum natura libros commentarius*. Impensis G. Reimeri, Berolini.
- Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Brodley, C. E. and Danyluk, A. P., editors, *Machine Learning International Workshop then Conference*, number Icml in ICML '01, pages 282–289. Citeseer.
- Laird, A. (2002). Juan Luis de la Cerda and the Predicament of Commentary. In Gibson, R. K. and Kraus, C. S., editors, *The Classical Commentary: Histories, Practices, Theory*, pages 171–203. Brill, Leiden.
- Le Boeuf, P. (2012). Modeling Rare and Unique Documents: Using FRBROO/CIDOC CRM. *Journal of Archival Organization*, 10(2):96–106.
- Le Boeuf, P. (2012). A Strange Model Named FRBROO. *Cataloging & Classification Quarterly*, 50(5-7):422–438.
- Lee, J. (2007). A Computational Model of Text Reuse in Ancient Literary Texts. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 472–479, Prague, Czech Republic. Association for Computational Linguistics.
- Lopez, P. (2009). GROBID: Combining Automatic Bibliographic Data Recognition and Term Extraction for Scholarship Publications. In *Proceedings of the 13th European Conference on Research and Advanced Technology for Digital Libraries, ECDL'09*, pages 473–474, Berlin, Heidelberg. Springer-Verlag.
- Manca, M., Spinazzè, L., Mastandrea, P., and Tessarolo, L. (2011). Musique Deoque : Text Retrieval on Critical Editions. *Journal for Language Technology and Computational Linguistics*, 26(2):129–140.
- Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

- Mastandrea, P. and Spinazzè, L. (2011). *Nuovi archivi e mezzi d'analisi per i testi poetici: i lavori del progetto Musisque Deoque, Venezia, 21-23 giugno 2010*. Adolf Hakkert, Amsterdam.
- Mayfield, J., McNamee, P., and Piatko, C. (2003). Named Entity Recognition Using Hundreds of Thousands of Features. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4, CONLL '03*, pages 184–187, Stroudsburg, PA, USA. Association for Computational Linguistics.
- McCarty, W. (2002). A Network with a Thousand Entrances: Commentary in an Electronic Age? In Gibson, R. K. and Kraus, C. S., editors, *The Classical Commentary: Histories, Practices, Theory*, Mnemosyne Supplements, pages 359–402.
- McCarty, W. (2004). Modeling: A Study in Words and Meanings. In *A Companion to Digital Humanities*, pages 254–270. Blackwell Publishing Professional, Oxford.
- McCarty, W. (2013). What does Turing have to do with Busa? In Mambrini, F., Passarotti, M., and Sporleder, C., editors, *Proceedings of The Third Workshop on Annotation of Corpora for Research in the Humanities*, pages 1–14.
- McCarty, W. (2014). Special Effects; or, The Tooling Is Here. Where Are the Results? In Dershowitz, N. and Nissan, E., editors, *Language, Culture, Computation. Computing of the Humanities, Law, and Narratives*, number 8002 in Lecture Notes in Computer Science, pages 103–117. Springer Berlin Heidelberg.
- McDonough, J. (1967). Computers and the classics. *Computers and the Humanities*, 2(1):37–40.
- Meho, L. I. and Tibbo, H. R. (2003). Modeling the information-seeking behavior of social scientists: Ellis's study revisited. *Journal of the American Society for Information Science and Technology*, 54(6):570–587.
- Mizoguchi, R. (2004). Tutorial on Ontological Engineering Part 3: Advanced Course of Ontological Engineering. *New Generation Computing*, 22(2):193–220.
- Moretti, F. (2007). *Graphs, maps, trees : abstract models for a literary history*. Verso, London.

- Murai, H. and Tokosumi, A. (2005). A Network Analysis of Hermeneutic Documents Based on Bible Citations. In Bara, B. G., Lawrence Barsalou, and Bucciarelli, M., editors, *CogSci 2005*, pages 1565–1570.
- Murai, H., Tokosumi, A., Tokunaga, T., and Ortega, A. (2008). Extracting concepts from religious knowledge resources and constructing classic analysis systems. In Tokunaga, T. and Ortega, A., editors, *Large-Scale Knowledge Resources. Construction*, volume 4938 of *Lecture Notes in Computer Science*, pages 51–58, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26.
- Newman, M. (2010). *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA.
- Odat, S., Groza, T., and Hunter, J. (2015). Extracting structured data from publications in the Art Conservation Domain. *Literary and Linguistic Computing*, 30(2):225–245.
- O'Donnell, J. J. (1998). *Avatars of the word : from papyrus to cyberspace*. Harvard University Press, Cambridge, Mass.
- Paijmans, H. and Wubben, S. (2007). Preparing Archaeological Reports for Intelligent Retrieval. In *Proceedings of the 35th Annual Conference on Computer Applications and Quantitative Methods in Archaeology (CAA2007)*, pages 212–217, Berlin, Germany.
- Palmer, C., Tefteau, L., and Pirmann, C. (2009). Scholarly Information Practices in the Online Environment: Themes from the Literature and Implications for Library Service Development.
- Parr, T. J. and Quong, R. W. (1995). ANTLR: A predicated-LL(k) parser generator. *Software: Practice and Experience*, 25(7):789–810.
- Pease, A. S. (1935). *Publi Vergili Maronis Aeneidos: liber quartus*. Harvard University Press, Cambridge, Mass.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and

- Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Peroni, S. and Shotton, D. (2012). FaBiO and CiTO: Ontologies for describing bibliographic resources and citations. *Web Semantics: Science, Services and Agents on the World Wide Web*, 17:33–43.
- Porter, D., Du Casse, W., Jaromczyk, J. W., Moore, N., Scaife, R., and Mitchell, J. (2006). Creating CTS collections. In *Proceedings of Digital Humanities 2006*, pages 269–274, Paris.
- Presner, T. (2012). How to Evaluate Digital Scholarship. <http://journalofdigitalhumanities.org/1-4/how-to-evaluate-digital-scholarship-by-todd-presner/>.
- Presutti, V. and Gangemi, A. (2008). Content Ontology Design Patterns as Practical Building Blocks for Web Ontologies. In *Conceptual Modelling - ER 2008*, volume 5231, pages 128–141.
- Radicchi, F., Fortunato, S., and Vespignani, A. (2012). Citation Networks. In Scharnhorst, A., Börner, K., and van den Besselaar, P., editors, *Models of Science Dynamics, Understanding Complex Systems*, pages 233–257. Springer Berlin Heidelberg.
- Richardson, J. (2005). Indexing Roman imperialism. *The Indexer*, 24(3):138–140.
- Richardson, J. (2008). *The language of empire: Rome and the idea of empire from the third century BC to the second century AD*. Cambridge University Press, Cambridge, UK; New York.
- Rochat, Y. (2014). Character Networks and Centrality.
- Romanello, M. (2013). Creating an Annotated Corpus for Extracting Canonical Citations from Classics-Related Texts by Using Active Annotation. In Gelbukh, A., editor, *Computational Linguistics and Intelligent Text Processing. 14th International Conference, CICLing 2013, Samos, Greece, March 24-30, 2013, Proceedings, Part I*, volume 1 of *Lecture Notes in Computer Science / Theoretical Computer Science and General Issues*, pages 60–76. Springer Berlin Heidelberg.

- Romanello, M., Berti, M., Babeu, A., and Crane, G. (2009a). When printed hypertexts go digital: information extraction from the parsing of indices. pages 357–358, Torino, Italy. ACM.
- Romanello, M., Berti, M., Boschetti, F., Babeu, A., and Crane, G. (2009b). Rethinking Critical Editions of Fragmentary Texts By Ontologies. In Mornati, S. and Hedlund, T., editors, *Rethinking Electronic Publishing: Innovation in Communication Paradigms and Technologies - Proceedings of the 13th International Conference on Electronic Publishing*, pages 155–174, Milano, Italy.
- Romanello, M., Boschetti, F., and Crane, G. (2009c). Citations in the digital library of classics: extracting canonical references by using conditional random fields. In *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, NLP4DL '09, pages 80–87, Morristown, NJ, USA. Association for Computational Linguistics.
- Romanello, M. and Pasin, M. (2011). An Ontological View of Canonical Citations. In *Digital humanities 2011: conference abstracts*, pages 216–218, Stanford. Stanford University Library.
- Romanello, M. and Pasin, M. (2013). Citations and Annotations in Classics : Old Problems and New Perspectives. In *DH-CASE '13 Proceedings of the 1st International Workshop on Collaborative Annotations in Shared Environment: metadata, vocabularies and techniques in the Digital Humanities*, New York, NY, USA. ACM.
- Romanello, M. and Thomas, A. (2012). The World of Thucydides: From Texts to Artefacts and Back. In Zhou, M., Romanowska, I., Zhongke, W., Pengfei, X., and Verhagen, P., editors, *Revive the Past. Proceeding of the 39th Conference on Computer Applications and Quantitative Methods in Archaeology. Beijing, 12-16 April 2011*, pages 276–284. Amsterdam University Press.
- Roueché, C., Lawrence, K., and Lawrence, K. F. (2014). Linked Data and Ancient Wisdom. *ISAW Papers*, 7(25).
- Ruhleder, K. (1995). Reconstructing Artifacts, Reconstructing Work: From Textual Edition to On-Line Databank. *Science, Technology, & Human Values*, 20(1):39–64.

- Rydberg-Cox, J. A. (2006). *Digital libraries and the challenges of digital humanities*. Chandos Pub., Oxford.
- Scaife, R. (2006). False multiples in the TLG Canon. <http://www.stoa.org/archives/330>.
- Schich, M. and Coscia, M. (2011). Exploring Co-Occurrence on a Meso and Global Level Using Network Analysis and Rule Mining. In *Proceedings of the ninth workshop on mining and Learning with Graphs MLG 11*. ACM.
- Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49.
- Schmid, H. (1995). Improvements In Part-of-Speech Tagging With an Application To German. In *Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.
- Seifert, S. (2014a). Brief von August Boeckh an Karl August Varnhagen von Ense (Berlin, 29. März 1844). <http://tei.ibi.hu-berlin.de/berliner-intellektuelle/manuscript?Brief109BoeckhanVarnhagen>.
- Seifert, S. (2014b). Brief von August Boeckh an Karl August Varnhagen von Ense (Berlin, 30. November 1845). <http://tei.ibi.hu-berlin.de/berliner-intellektuelle/manuscript?Brief110BoeckhanVarnhagen>.
- Sellars, J. (2013). Some Sixteenth-Century Editions of Ancient Philosophical Texts in Wolfson College Library. *The Bodleian Library Record*, 26(1):94–102.
- Settles, B. (2004). Biomedical named entity recognition using conditional random fields and rich feature sets. In *JNLPBA '04: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 104–107, Morristown, NJ, USA. Association for Computational Linguistics.
- Settles, B. (2009). Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

- Shadbolt, N., Hall, W., and Berners-Lee, T. (2006). The Semantic Web Revisited. *Intelligent Systems, IEEE*, 21(3):96–101.
- Shotton, D. (2009). Semantic publishing: the coming revolution in scientific journal publishing. *Learned Publishing*, 22(2):85–94.
- Shotton, D. (2010). CiTO, the Citation Typing Ontology. *Journal of biomedical semantics*, 1(Suppl 1):S6.
- Shotton, D., Portwin, K., Klyne, G., and Miles, A. (2009). Adventures in Semantic Publishing: Exemplar Semantic Enhancements of a Research Article.
- Silvio Peroni, Alexander Dutton, Tanya Gray, and David Shotton (2015). Setting our bibliographic references free: towards open citation data. *Journal of Documentation*.
- Simon, R., Barker, E., de Soto, P., and Isaksen, L. (2014). Pelagios. *ISAW Papers*, 7(27).
- Simon, R., Barker, E., and Isaksen, L. (2012). Exploring Pelagios: a visual browser for geo-tagged datasets. In Agirre, E., Fernie, K., Otegi, A., and Stevenson, M., editors, *International Workshop on Supporting Users' Exploration of Digital Libraries*, pages 29–34.
- Smith, B. (2003). Ontology. In Floridi, L., editor, *Blackwell Guide to the Philosophy of Computing and Information*, pages 155–166. Blackwell, Oxford.
- Smith, N. (2009). Citation in Classical Studies. *Digital Humanities Quarterly*, 3(1).
- Smith, N. (2010). Digital Infrastructure and the Homer Multitext Project. In Bodard, G. and Mahony, S., editors, *Digital Research in the Study of Classical Antiquity*, pages 121–137. Ashgate Publishing, Burlington, VT.
- Smolenaars, J. J. L. (2001). Statius Theb. 2.496-523: The Poet at Work. In Orbán, A. P. and van der Poel, M. G. M., editors, *Ad litteras: Latin studies in honour of J.H. Brouwers*, pages 241–257. Nijmegen University Press, Nijmegen.

- Sosin, J. (2014). Searching the DDbDP (Or, How Fine are a Balrog's Teeth?). <http://blogs.library.duke.edu/dcthree/2014/02/11/how-fine-are-a-balrogs-teeth/>.
- Staab, S. and Studer, R., editors (2009). *Handbook on Ontologies*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J. (2012). brat: a Web-based Tool for NLP-Assisted Text Annotation. In *Proceedings of the Demonstrations Session at EACL 2012*, Avignon, France. Association for Computational Linguistics.
- Stephens, S. (2002). Commenting on Fragments. In Gibson, R. K. and Kraus, C. S., editors, *The classical commentary : histories, practices, theory*, pages 67–88.
- Stone, S. (1982). Humanities Scholars: Information Needs and Uses. *Journal of Documentation*, 38(4):292–313.
- Sula, C. A. (2012). Visualizing social connections in the humanities: Beyond bibliometrics. *Bulletin of the American Society for Information Science and Technology*, 38(4):31–35.
- Sula, C. A. and Miller, M. (2014). Citations, contexts, and humanistic discourse: Toward automatic extraction and classification. *Literary and Linguistic Computing*, 29(3):452–464.
- Surdeanu, M., Foster, I. K., Rydholm, C. L., Nallapati, R. M., Walker, J. H., Gregory, G. D., Carothers, G., and Pilon, N. O. P. (2014). Systems and Methods for Using Non-Textual Information In Analyzing Patent Matters.
- Sure, Y., Staab, S., and Studer, R. (2009). Ontology Engineering Methodology. In *Handbook on Ontologies, International Handbooks*, pages 135–152. Springer-Verlag, Berlin, Heidelberg.
- Sutton, C. and McCallum, A. (2006). An Introduction to Conditional Random Fields for Relational Learning. In Getoor, L. and Taskar, B., editors, *Introduction to Statistical Relational Learning*, number x. MIT Press.

- Tibbo, H. R. (1993). *Abstracting, Information Retrieval, and the Humanities: Providing Access to Historical Literature*. Number no. 48 in {ACRL} publications in librarianship. American Library Association, Chicago.
- Trachsel, A. (2012). Collecting Fragments Today: What Status Will a Fragment Have in the Era of Digital Philology? In Clivaz, C., Meizoz, J., Vallotton, F., and Verheyden, J., editors, *Lire demain - Reading tomorrow*, pages 415–429. Presses polytechniques et universitaires romandes (PPUR), Lausanne.
- Tupman, C., Hedges, M., Jordanous, A., Roueche, C., Lawrence, K. F., Wakelnig, E., and Dunn, S. (2012). Sharing Ancient Wisdoms: developing structures for tracking cultural dynamics by linking moral and philosophical anthologies with their source and recipient texts. In *Digital Humanities conference, Hamburg, Germany*.
- Unsworth, J. (2000). Scholarly Primitives: what methods do humanities researchers have in common, and how might our tools reflect this? <http://people.brandeis.edu/~unsworth/Kings.5-00/primitives.html>.
- Vandendorpe (2009). *From papyrus to hypertext : toward the universal digital library*. University of Illinois Press, Urbana.
- Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA.
- Vlachos, A. (2006). Active annotation. *Proceedings of the Workshop on Adaptive Text Extraction and Mining (ATEM 2006)*, pages 64–71.
- von Wilamowitz-Moellendorff, U. (1895). *Euripides Herakles Erklärt von Ulrich von Wilamowitz-Moellendorff*. Weidmann, Berlin.
- Waite, S. (1970). Computers and the classics. *Computers and the Humanities*, 5(1):47–51.
- Watson-Boone, R. (1994). The Information Needs and Habits of Humanities Scholars. *RQ*, 34(2):203–215.
- Weingart, S. (2014). Submissions to Digital Humanities 2015 (pt. 2). <http://www.scottbot.net/HIAL/?p=41053>.
- Weingart, S. B. (2011). Demystifying Networks, Parts I & II. *Journal of Digital Humanities*, 1(1):15–33.

- Wiberley, S. E. and Jones, W. G. (1989). Patterns of Information Seeking in the Humanities. *College & Research Libraries*, 50(6):638–645.
- Wiberley Jr., S. E. (2009). *Humanities Literatures and Their Users*, pages 2197–2204. third edit edition.
- Willer, M. and Dunsire, G. (2013). *Bibliographic Information Organization In The Semantic Web*. Elsevier Science, Burlington, VT.
- Winter, T. (1999). Roberto Busa, S.J., and the Invention of the Machine-Generated Concordance. *The Classical Bulletin*, 75(1):3–20.
- Wong, W., Liu, W., and Bennamoun, M. (2012). Ontology Learning from Text: A Look Back and into the Future. *ACM Comput. Surv.*, 44(4):20:1–20:36.
- Zola, N. J. (2012). Why are there verses missing from my Bible? The emergence of verse numbers in the New Testament. *Restoration Quarterly*, 54(4):241–253.